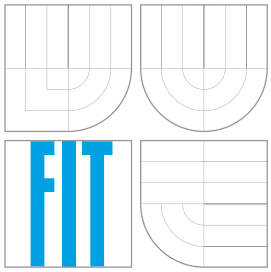
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ZABEZPEČENÝ TRANSPORTNÍ PROTOKOL MONITOROVACÍCH SYSTÉMŮ

SECURE TRANSPORT PROTOCOL FOR SYSTEM MONITORING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PATRIK HALFAR

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2010

Abstrakt

Dokument se zabývá analýzou NetFlow protokolů a řešením jeho nedostatků. Prioritně je zaměřen na důvěrnost a spolehlivost přenosu s respektováním dodržení minimálních nároků na výpočetní zdroje. Diskutovaná řešení uvažují skutečnost, že nelze měnit činnost exportéru mimo jím definované parametry.

Abstract

This paper deals with NetFlow protocol and analysis of its properties. There is focused on confidentiality and reliability of the data's transfer. All of the solution observe requirements for lower resources of devices, and impossibility of changes in the exporter except defined parameters.

Klíčová slova

šifrování, IP protokol, bezstavový protokol, NetFlow

Keywords

cryptography, IP protocol, stateless protocol, NetFlow

Citace

Patrik Halfar: Zabezpečený transportní protokol monitorovacích systémů, diplomová práce, Brno, FIT VUT v Brně, 2010

Zabezpečený transportní protokol monitorovacích systémů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Matouška, Ph.D.

.....
Patrik Halfar
23. května 2010

Poděkování

Děkuji Ing. Petru Matouškovi, PhD. za vedení diplomové práce a cenné připomínky, Ing. Matěji Grégrovi a Ing. Tomáši Podermańskimu za odborné konzultace .

Obsah

Úvod

Počítačové sítě patří v dnešní době mezi nejvíce využívané médium pro sdílení informací a poskytování služeb. Spravovat počítačovou síť vyžaduje znát její zapojení, vlastnosti, způsob jakým funguje a jak je nastavena. Důležité je znát její vytížení, aktuální stav, jak přes síť procházejí informace, kde jsou její limity nebo slabá místa. To vše z pohledu zatížení tak i z hlediska bezpečnostního. Dávno jsou pryč doby, kdy stačilo na lince vedoucí do veřejné sítě mít nastaven firewall a věřit, že síť je ochráněna. Dnešní útoky využívají sofistikovaných přístupů, aby umožnily obejít různé stupně zabezpečení.

Monitorování sítě je jednou z možností, jak zvýšit její stabilitu, propustnost i odolnost. Mnoho informací o síti můžeme zjistit například pomocí protokolu *SNMP* (Simple Network Managment Protocol). Ve většině případech se jedná o monitorování stavu sítě (čítače, směrovací tabulky, apod.). Z těchto informací dokážeme zjistit, že je síť přetížena. Bohužel data neposkytnou příčinu přetížení. Aby bylo možné zjistit příčiny zahlcení sítě, je nezbytně nutné mít informace o tom, jaká data sítí protékají (ne nezbytně jejich obsah, stačí meta data – typ služby, odkud, kam, atd.).

V otázce bezpečnosti může monitorování síťového provozu poskytnout široké možnosti pro odhalování cílených útoků. Stejně tak ho lze využít i pro detekci škodlivého software, ať už je jeho primárním účelem napadání síťových služeb nebo síť využívá jen pro své šíření.

> *Útok na síť vždy začíná zkoumáním sítě (network reconnaissance), kdy se útočník pokouší o neautorizované zmapování sítě, služeb nebo zranitelností. Cílem této fáze je získat dostatek informací k provedení dalších typů útoků se zaměřením na získaní přístupu nebo odmítnutí přístupu legitimním uživatelům. Pokud by byla počáteční fáze (tedy zkoumání) neúspěšná, další útok by byl pro útočníka daleko obtížněji realizovatelný. [9]*

Jsou-li uvedené informace k dispozici, pak mohou najít i další uplatnění, například mohou být využity ke zpoplatnění služeb, monitorování zaměstnanců, zda v pracovní době nenavštěvují internetové stránky z nežádoucích zdrojů, aj. Otázkou je, zda-li existence těchto údajů nepředstavuje riziko pro uživatele sítě a zda nemohou být zneužity proti nim. To, že se uživatel připojuje k nějakému serveru, nemusí být protizákonné, ale zveřejnění takové informace by ho mohlo přesto poškodit. V této souvislosti je nutné mít jistotu, že získávají-li se taková data, pak jsou dostatečně zabezpečena před zneužitím.

Jednou z možností, jak výše popsané údaje získávat, je *NetFlow*. Data získána pomocí něj se obvykle používají k jednomu ze tří účelů:

- statistické údaje o provozu – často anonymizovaná, obvykle nepředstavují hrozbu pro uživatele, problém je v tom, že se anonymizují až v kolektoru, proto je nutné mít jistotu, že nebyla zneužita ještě před anonymizací,
- účtování – přesné informace o spotřebovaných zdrojích sítě,

- analýza sítě – umožní odhalit slabá nebo předimenzovaná místa, vytvořit matematický model zatížení sítě.

Z výše uvedeného vyplývá, že jsou sbírána data s různou citlivostí a důležitostí. Proto je tato diplomová práce zaměřena na analýzu možností pro zajištění bezpečného přenosu *NetFlow* dat. Získaná data budou využita pro návrh řešení, které dokáže zajistit důvěrný a bezztrátový přenos dat mezi exportérem a kolektorem přes veřejnou síť.

Cíl

Cílem této práce je návrh způsobu pro zajištění důvěrného a spolehlivého přenosu dat z exportéru do kolektoru přes veřejnou počítačovou síť. Zároveň musí být respektována všechna omezení vyplývající z realizace exportéru v aktivních prvcích nebo ve vestavěných systémech. Navržené řešení by mělo mít snadné nasazení a správu, která umožní snadné začlenění do existujících instalací. Prvotní impuls pro řešení této problematiky vzešel z Centra výpočetních a informačních služeb Vysokého učení technického v Brně, kde se předpokládá nasazení poznatků nalezených v rámci této práce. Tento fakt může ovlivnit poznatky v tom rozsahu, že pokud nebude možné nalézt „univerzální“ řešení, pak se uplatní takové, které bude splňovat požadavky tohoto centra.

Členění práce

První část práce je zaměřena na popis vlastností protokolu NetFlow a dalších alternativních protokolů, které jsou navrženy pro monitorování provozu na síti. Součástí je i analýza vlivu na sítě s vysokou dostupností a zajištění této služby pro monitorovací systém. Druhá část práce je zaměřena na analýzu možností jak zabezpečit transport dat protokolu NetFlow a popis realizace navrženého řešení včetně testování.

Kapitola 1 – Rozbor protokolu NetFlow

Kapitola 2 – Rozbor protokolu IPFIX

Kapitola 3 – Rozbor protokolu sFlow

Kapitola 4 – Analýza slabých míst

Kapitola 5 – Srovnání protokolů

Kapitola 6 – Rozbor možností pro realizaci zabezpečení

Kapitola 7 – Popis realizace, nasazení a testování navrženého řešení

Kapitola 1

NetFlow

Tato kapitola je zaměřena na popis NetFlow. Protokol vznikl v dílně firmy Cisco Systems. NetFlow je služba umožňující sbírat metadata o provozu sítě ve formě „telefonního účtů". Informace, které NetFlow poskytuje, nabízí odpovědi na otázky, které se typicky vážou k účtu za telefon [14]:

- Kdo s kým komunikoval?
- Kdy komunikace proběhla?
- Jak dlouho trvala?
- Jakou službou (protokol a port)?
- Kolik dat bylo přeneseno?

Odpovědi na tyto otázky lze získat z paměti NetFlow záznamů v aktivních prvcích. Už při vzniku této služby její návrháři definovali tři oblasti, ve kterých NetFlow nalézá uplatnění:

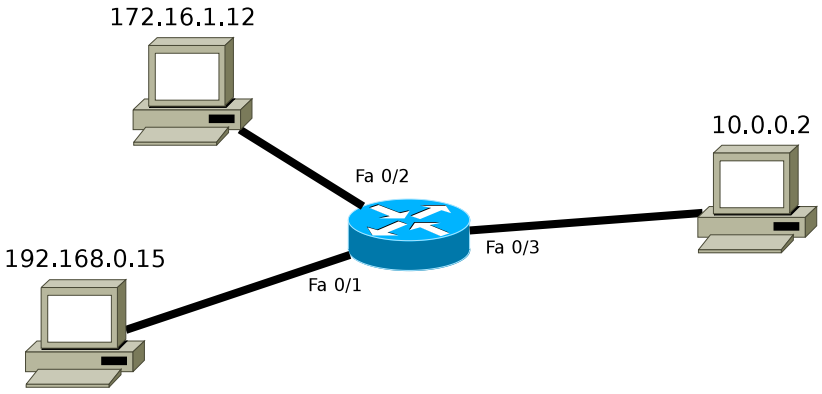
- účtování,
- analýza vytížení sítě,
- analýza útoků pomocí hledání anomálií.

Z pohledu NetFlow je jednotkou komunikace *tok*. Na rozdíl od telefonního hovoru je tok definován pro každý směr samostatně. Tato vlastnost vychází ze samotné podstaty sítí založených na přepínaní paketu, neboť odpověď může procházet jinou trasou než požadavek. U telefonních hovorů, které se uskutečňují formou spojování okruhu, jde vždy komunikace pro oba směry stejnou trasou. Každý tok je tedy identifikován sedmicí (zdrojová adresa, zdrojový port, cílová adresa, cílový port, protokol, ToS[1], vstupní rozhraní). Hodnota protokol identifikuje protokol použitý na vyšší vrstvě. Obvyklé hodnoty jsou 1 – ICMP, 6 – TCP, 17 – UDP. Seznam přiřazených hodnot eviduje IANA (Internet Assigned Numbers Authority) [2]. Příklad identifikace toků je znázorněn na obrázku 1.1.

1.1 Exportér

Datový tok vzniká v exportéru. Původně se nacházel jako softwarový agent na aktivním prvku (obr. 1.2), který zajišťuje směrování. Tato zařízení obsahují potřebná data v paměti

[1] Type of Service, původní účel se nikdy neuplatnil, dnes se používá ve spojení s QoS.



Zdrojová adresa	Zdrojový port	Cílová adresa	Cílový port	Protokol	ToS	Vstupní rozhraní
192.168.0.15	1234	172.16.1.12	80	6	40	Fa 0/1
172.16.1.12	80	192.168.0.15	1234	6	40	Fa 0/2
192.168.0.15	2222	10.0.0.2	53	17	80	Fa0/1

Obrázek 1.1: Identifikace toků

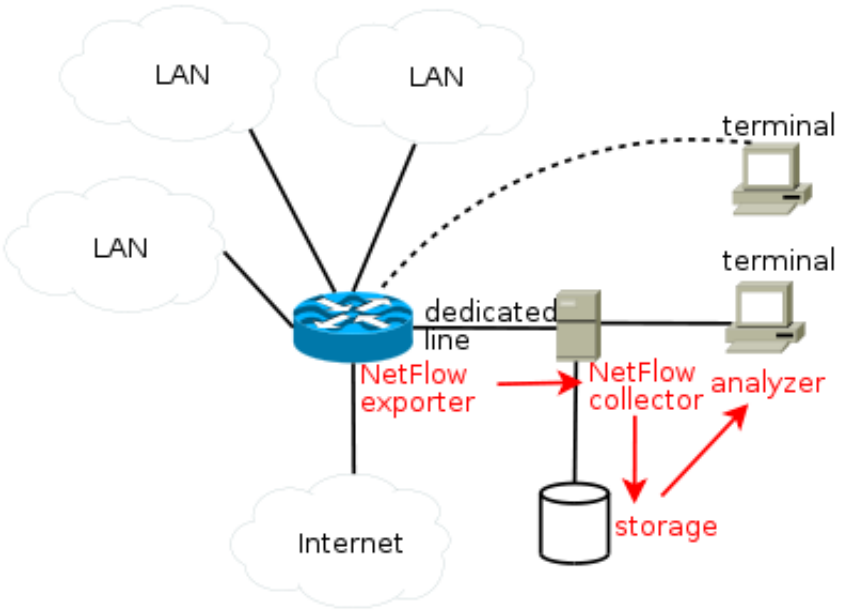
pro účely rychlejšího směrování. Rozšířením této paměti o některé další položky se získá paměť se záznamy NetFlow(tab. 1.1). Zvyšováním rychlostí linek se zvyšují požadavky na primární činnost routerů, resp. L3 přepínačů, tedy směrování. To znamená, že sekundární služby musí ustoupit a tyto zařízení je mohou vykonávat jen pokud mají volné prostředky. Důsledkem je, že toky, které se zaznamenávají na aktivních prvcích mohou sloužit pro statistickou analýzu, kde je přípustné získávat jen každý n-tý vzorek bez většího ovlivnění statistiky. V žádném případě nemohou být podkladem pro účtování. Účinnost pro odhalení útoků se rovněž snižuje.

SrcIf	**SrcIPadd**	**DstIf**	**DstIPadd**	**Protocol**	**TOS**	**Flgs**	**Pkts**	**SrcPort**
Fa 1/0	173.100.21.2	Fa 0/0	10.0.227.12	11	80	10	11000	162
Fa 1/0	173.100.3.2	Fa 0/0	10.0.227.12	6	40	0	2491	15
Fa 1/0	173.100.20.2	Fa 0/0	10.0.227.12	11	80	10	10000	161
Fa 1/0	173.100.6.2	Fa 0/0	10.0.227.12	6	40	0	2210	19

SrcMsk	**SrcAS**	**DstPort**	**DstMsk**	**DstAS**	**NextHop**	**Byts**/Pkt	**Active**	**Idle**
/24	5	163	/24	15	10.0.23.2	1528	*1745*	4
/25	196	15	/24	15	10.0.23.2	740	41.5	1
/24	180	10	/24	15	10.0.23.2	1428	1145.5	3
/30	180	19	/24	15	10.0.23.2	1040	24.5	*14*

Tabulka 1.1: Příklad NetFlow cache [1]

Řešením problému mohou být přídavné moduly do aktivních prvků vyhrazené pro podporu monitorování toků na síti. Další možností je použití samostatných zařízení, která se do sítě připojí pomocí replikace portu (SPAN – Switch Port Analyzer) nebo speciálního zařízení Tap (viz dále). Uvedená zařízení mohou být realizována pomocí speciálního hardware, ovšem častěji se objevují realizace formou počítače s předinstalovaným softwarem. Výhoda těchto zařízení spočívá mimo jiné také v tom, že je lze nasadit i v rámci podsítě, kde není provoz směrován, ale pouze přepínán. Taková zařízení se nazývají sondy. Příklad zapojení sond je vyobrazen na obr. 1.3. Informace o dokončených datových tocích jsou z exportéru přenášeny do kolektoru a odstraněny z paměti.



Obrázek 1.2: NetFlow architektura: exportér v aktivním prvku [24]

Tap

Tap, neboli test access port, je pasivní zařízení, které se připojuje na linku mezi dvě zařízení (monitorované porty) a na další rozhraní (monitorovací) duplikuje provoz, který probíhá mezi monitorovanými rozhraními. Popsané zařízení neovlivňuje provoz na monitorované lince a to ani v případě výpadku napájení Tap.

1.2 Kolektor

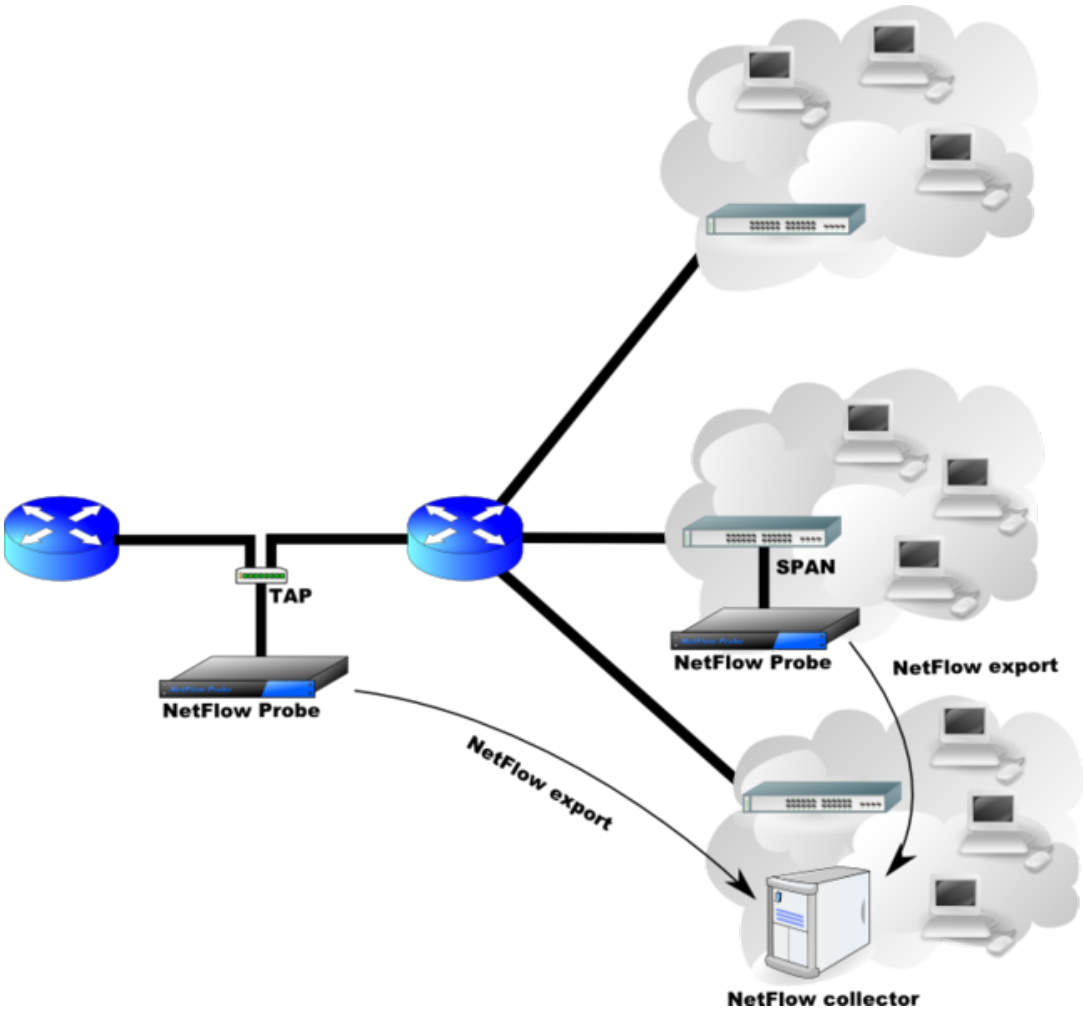
Cílem kolektoru je shromažďovaní dat, která vyprodukují exportéry. Navržený koncept umožňuje koncept $M : N$. Je tedy možno exportovat data z M exportéru na více než jeden kolektor. V praxi je obvyklé, že $N = 1$, nebo že kolektor je spojen v jedno zařízení s exportérem. Pak se data zpracovávají přímo na sondě a výstup je obvykle dostupný prostřednictvím webového rozhraní.

1.3 Životní cyklus toku

Nový datový tok vznikne příchodem paketu, který nelze asociovat s žádným již existujícím tokem. Tedy neexistuje-li záznam, který by měl všech sedm parametrů shodných. Vznik nového toku znamená vytvoření záznamu v paměti, inicializaci čítačů a přiřazení časové značky.

Každý následující paket, který je možné asociovat s existujícím tokem, znamená navýšení čítačů (počtu prošlých paketů, velikosti přenesených dat, apod.)

Ukončení datového toku již nelze exaktně definovat. Pokud jde o spojově orientovaný přenos, pak ukončení spojení signalizuje nastavení příznaku FIN nebo RST v paketu. V případě výpadku spojení se takový paket neobjeví. Takový tok by tedy nebyl nikdy ukončen. Další problém nastává v případě dlouhých spojení. Například nějaká aplikace je připojena



Obrázek 1.3: NetFlow architektura: externí sondy [24]

k databázi, od které se odpojí jen v případě nějaké chyby. Aby tyto situace nenastávaly, může být tok ukončen z důvodu

- neaktivního spojení (*pasivní timeout*) – delší dobu nebyl zaznamenán žádný paket,
- dlouhotrvající spojení (*aktivní timeout*) – tok již trvá déle než je nastavená maximální délka toku,
- není prostor pro uložení nového toku – nejstarší toky se označí za ukončené,
- přišel paket s příznakem FIN nebo RST.

Z ukončených toků vzniká *NetFlow záznam* (Flow Record). Struktura záznamu je závislá na verzi. Příklad nejpoužívanější verze (verze pět) je zobrazena v tabulce 1.5. Prvních osm verzí je proprietárních, kdežto verze devátá byla zveřejněna v RFC 3954 [6].

Zpracování paketu pro úpravu záznamu o toku v NetFlow se provádí procesorem, proto při velkém množství paketu může vzniknout situace, že exportér nebude stíhat. Aby se takovéto situaci předešlo, může být kolektor nastaven tak, že bude provádět vzorkování. V takovém případě se nezpracovává každý paket, ale definuje se vzorkovací perioda.

1.4 Exportovaná data

Komunikace protokolem NetFlow mezi exportérem a kolektorem probíhá bezstavově a pouze jediným směrem. Výhodou tohoto způsobu exportu jsou nízké požadavky na procesorový čas a paměť v exportéru. Tento způsob doručování zůstává zachován bez ohledu na verzi protokolu. Jak již bylo dříve zmíněno, nejrozšířenější je pátá verze protokolu a obvykle spolu s verzí devět jsou pouze tyto dvě podporovány. Rozdíly mezi verzemi jsou zobrazeny v tabulce 1.2.

Verze	Popis	Standard
1	První implementace, omezení na IPv4, bez masky sítě	Proprietární
2	Nebyla uvolněna	Proprietární
3	Nebyla uvolněna	Proprietární
4	Nebyla uvolněna	Proprietární
5	V současnosti nejrozšířenější, omezena na IPv4	Proprietární
6	Podpora zapouzdřeného provozu (tunelování)	Proprietární
7	Získávání informací z přepínačů	Proprietární
8	Podpora agregace – pouze informací, které jsou obsaženy ve verzi pět	Proprietární
9	Šablony dat, podpora IPv6	RFC 3954

Tabulka 1.2: Přehled verzí NetFlow protokolu [24]

Zásadní změna, kterou přináší verze devět, je zrušení statické podoby paketu. Tato verze zavádí dva typy paketu:

- šablonu a
- data.

Šablona definuje sémantiku hodnot, které se budou přenášet. Data jsou pak tvořena trojicemi TLV (typ, délka, hodnota). Komplikaci do tohoto protokolu vnáší skutečnost, že šablona může přijít kdykoliv. To i několikrát v rámci jednoho paketu. Důležitým rozšířením, které přináší tato verze, je podpora IPv6.

Další změnou je, že devátá verze byla navržena tak, aby byla nezávislá na transportním protokolu a předpokládala se podpora transportního protokolu SCTP (Stream Control Transmission Protocol)[6].

1.5 Vlastnosti přenosu

Export NetFlow záznamů je realizován zasláním zprávy z exportéru do kolektoru. Formát zprávy pro verzi pět je uveden v tabulce 1.3. Z tabulky vyplývá, že data jsou přenášena v otevřené formě. Standard upravující protokol NetFlow, který byl uvolněn (verze devět) se o bezpečnosti nebo spolehlivosti přenosu nezmiňuje. Následující část je zaměřena na rozbor protokolu z hlediska důvěrnosti a spolehlivosti.

00	01	02	03	04	05	06	07
version		count		sys_uptime			
unix_secs				unix_nsecs			
flow_sequence				engine_type	engine_id	sampling_interval	

srcaddr				dstaddr			
nexthop				input		output	
dPkts				dOctets			
first				last			
srcport		dstport		pad1	tcp_flags	prot	tos
src_as		dst_as		src_mask	dst_mask	pad2	

Tabulka 1.3: Formát paketu NetFlow verze pět [22]

Spolehlivost

Jedním z důvodu nasazení NetFlow je účtování na základě počtu přenesených dat, spojení, apod. Otázkou tedy je, zda je přípustná ztráta paketu nesoucí informaci o uskutečněných tocích. Z pohledu uživatele je tato ztráta samozřejmě přípustná. Při ztrátě paketů je pravděpodobné, že nějaká služba, nebo její část nebude naúčtována. Z pohledu poskytovatele to představuje problém, neboť ztráta dat znamená ztrátu příjmů.

Jiným důvodem je získávání statistických dat pro úpravu návrhu sítě na základě informací o vytížení sítě. V takové situaci ztráta dat nepředstavuje problém, pokud by se jednalo jen o ztrátu několika málo paketů. Dojde-li ke ztrátě dat z důvodu přetížení linky, je žádoucí ztrátě dat zabránit, neboť data, která se ztrácejí mohou nést informace o příčinách přetížení.

Označení	**Význam**
version	Verze NetFlow protokolu
count	Počet toků exportovaných v tomto paketu (1-30)
sys_uptime	Čas v milisekundách od spuštění exportéru
unix_secs	Aktuální čas v sekundách od 1. 1. 1970 UTC
unix_nsecs	Aktuální čas v nanosekundách od 1. 1. 1970 UTC
flow_sequence	Celkový počet toků „spatřených" exportérem
engine_type	Typ slotu
engine_id	Identifikace slotu
sampling_interval	První dva bity identifikují režim vzorkování, dalších čtrnáct bitů vzorkovací interval

Tabulka 1.4: Význam polí v hlavičce paketu NetFlow verze pět [22]

Označení	Význam
srcaddr	Zdrojová IP adresa
dstaddr	Cílová IP address
nexthop	IP adresa, kam byl tok přeposílán
input	SNMP index vstupního rozhraní
output	SNMP index výstupního rozhraní
dPkts	Počet paketů v toku
dOctets	Celkový počet bytů přenesených v rámci toků na třetí vrstvě
first	Čas zachycení prvního paketu (začátek toku)
last	Čas zachycení posledního paketu (konec toku)
srcport	Zdrojový port
dstport	Cílový port
pad1	Nevyužité byty
tcp_flags	Příznaky paketů
prot	Protokol vyšší vrstvy (např. TCP = 6; UDP = 17)
tos	Type of Service
src_as	Autonomní systém zdroje
dst_as	Autonomní systém cíle
src_mask	Síťová maska zdrojové adresy
dst_mask	Síťová maska cílové adresy
pad2	Nevyužité byty

Tabulka 1.5: Význam datových polí paketu NetFlow verze pět [22]

Třetím důvodem pro nasazení NetFlow je bezpečnost. Má-li NetFlow sloužit k odhalení útoků, pak ztráta několika paketů nebude problémem, protože většina útoků se bude projevovat mnoha spojeními na různé cíle z jednoho místa. Pokud ovšem byla síť vystavena úspěšnému útoků, pak je žádoucí mít k dispozici všechna data, protože mohou pomoci identifikovat všechny kompromitované stroje a způsobené škody.

Komunikace mezi exportérem a kolektorem probíhá pouze jedním směrem. Exportér zasílá zprávy o tocích na kolektor. Kolektor nemá možnost komunikovat s exportérem. Pokud tedy dojde ke ztrátě zprávy, je tato ztráta perzistentní, neboť protokol UDP použitý na transportní vrstvě tuto vlastnost nezajišťuje a na vyšší vrstvě nemá kolektor možnost požádat o opětovný přenos nedoručených dat. V hlavičce paketu (tab. 1.3, 1.4) se kromě počtu toků přenášených v paketů uvádí i počet „spatřených" toků (*flow_sequence*). Tato hodnota umožňuje spočítat počet toků, o nichž data nebyla doručena (počet spatřených toků - součet všech přijatých toků). I když je tato vlastnost pozitivní, neposkytuje dostatečné zajištění pro všechny stanovené cíle.

Důvěrnost

Tok popisuje kdo s kým, kdy a jak dlouho komunikoval a pomocí jaké služby (informace není explicitně obsažena, ale lze odvodit z použitých portů a transportního protokolu. Z tohoto pohledu lze tato data označit za důvěrná a mělo by být s daty i takto zacházeno. Protokol NetFlow se důvěrností nezaobírá a ve standardu nejsou definována ani žádná doporučení. A cílem této práce je prozkoumat možnosti, jak tato data zabezpečit při přenosu mezi exportérem a kolektorem přes veřejnou síť.

1.6 Shrnutí

NetFlow byl prvním rozšířeným standardem pro monitorování dat a v dnešní době patří mezi nejrozšířenější. Za jeho vývojem a podporou stojí velká firma udávající směr v síťových technologiích. Důkazem jeho úspěšnosti je zajisté počet rozšíření, která jsou dostupná v různých verzích. Dalším důkazem je skutečnost, že se stal základem pro svého následníka IPFIX. Nicméně popis celého procesu sběru dat nikdy nebyl celý uvolněn a je definován jediným standardem, který definuje podobu exportovaných zpráv. Největším nedostatkem, se kterým se protokol potýká je jeho nezabezpečení.

Kapitola 2

IPFIX

Cílem diplomové práce je zajistit bezpečný transport pomocí protokolu NetFlow. V této a následující kapitole jsou popsány další dva protokoly, které slouží pro monitorování provozu na sítí stejně jako protokol NetFlow a poznatky zjištěné v těchto protokolech mohou být vodítkem pro dosažení bezpečného a spolehlivého přenosu dat v protokolu NetFlow.

IP Flow Information Export (IPFIX) [17], velmi často též označován jako desátá verze NetFlow, je plnohodnotným standardem, který definuje sběr údajů o tocích jako celek, včetně požadavků na činnost exportéru, přenosu i kolektoru. Návrh tohoto protokolu byl značně ovlivněn devátou verzí protokolu NetFlow. To může být způsobeno tím, že na jeho návrhu se podíleli i autoři NetFlow. Požadavků, které byly pro IPFIX definovány je mnoho. Neobvyklým krokem byl návrh, aby výchozím protokolem na čtvrté vrstvě ISO/OSI modelu pro přenos IPFIX záznamů byl protokol SCTP. A podpora klasických transportních protokolů TCP a UDP.

Implementace SCTP, včetně rozšíření PR-SCTP (upravené v RFC 3758 [20]), musí být implementováno ve všech verzích [3]. Podpora protokolu UDP a protokolu TCP je volitelná [5].

Rozšíření PR (Partial Reliability) protokolu SCTP slouží pro definici částečné spolehlivosti. Toto rozšíření dovoluje definovat časovou platnost zprávy. Pokud se zprávu nepodaří doručit v době platnosti, neprovádí se další pokus o doručení a zpráva může být tedy ztracena. Na rozdíl od protokolu UDP je zajištěno doručení ve správném pořadí.

2.1 Spolehlivost

Podpora SCTP a TCP přináší zásadní změnu v přístupu, protože znamená zavedení zpětného kanálu v komunikaci mezi exportérem a kolektorem. K protokolu UDP se standart IPFIX vyjadřuje:

> *UDP je nespolehlivý transportní protokol a negarantuje doručení zprávy a přenášená data mohou být tedy zatracena. UDP protokol nesmí být použit v aplikacích, které netolerují ztrátu dat. [5]*

Pro aplikace, kde je UDP protokol nepřijatelný, nabízí řešení zbývající dva protokoly (SCTP a TCP). Tyto protokoly implicitně zajišťují spolehlivý přenos a není nutné ho řešit na vyšší vrstvě. V dokumentu na požadované vlastnosti IPFIX (Requirements for IP Flow Information Export, RFC 3917 [17]) se vyžaduje, aby ztráta dat byla minimálně odhalitelná. V případě protokolu UDP je tedy požadavkem řešení na vyšší vrstvě, což v důsledku znamená zavedení sekvenčního čísla paketů, které umožní ztrátu detekovat.

2.2 Důvěrnost

IPFIX se také zabývá problémem důvěrnosti. Byly definovány tři bezpečnostní cíle, které musí být zajištěny:

- utajení,
- integrita,
- autentizace.

Pro zajištění těchto cílů se předpokládá použití protokolu TLS, resp. DTLS [18](DTLS je úpravou TLS pro protokol UDP a SCTP). Podpora protokolu DTLS musí být implementována, neboť protokol SCTP musí být povinně podporován [5].

Protokol DTLS (Datagram TLS) je úpravou protokolu TLS. Protokol TLS není možné použít při nasazení transportního protokolu, který negarantuje spolehlivý přenos a doručování paketu ve správném pořadí. V TLS není možné jednotlivé pakety dešifrovat nezávislé na sobě, ale je nezbytné zachovat kontext. Ztráta paketů zapříčiní nemožnost dešifrovat všechny následující zprávy. Doručování zpráv mimo pořadí vyžaduje skládání paketů na vyšší úrovni. V TLS na začátku spojení dochází k vyjednávaní parametrů a k autentizaci stran. Během této fáze jsou přenášená data, která mají typickou velikost několika kilobytů (teoretický až $2^{24} - 1$ bytů), kdežto velikost UDP datagramu je obvykle omezena na velikost 1500 bytů (maximální velikost ethernetového rámce [11]). DTLS používá pro úvodní vyjednávání stejné zprávy jako TLS, s tím že byly provedeny modifikace, aby v této fázi bylo možné zprávy fragmentovat a v případě ztráty paketu byl proveden opětovný přenos.

Dále je definován požadavek na podporu *push* a *pull* modelu. *Push* model je standardní chování, které je známe z NetFlow. Exportér zasílá zprávy na kolektor ve chvíli, kdy je nějaký tok ukončen. *Pull* model je definován jako volitelný. Jeho vlastností je, že umožňuje provést export dokončených toků externím spuštěním, tedy zasláním požadavků z kolektoru [17].

2.3 Shrnutí

Z výše uvedeného popisu je zřejmé, že autoři IPFIX v jeho návrhu zohlednili požadavky na utajovaný přenos. Otázka spolehlivosti je rovněž řešena a dovoluje uživateli výběr mezi spolehlivým a nespolehlivým přenosem volbou transportního protokolu. Protože IPFIX nabízí sběr stejných dat jako NetFlow rozšířený o zabezpečení spolehlivosti a důvěrnosti bylo by jeho nasazení řešením problému s NetFlow. Problém nasazení IPFIX je především v tom, že stále není standard zcela kompletní a podpora v zařízeních a softwaru není stoprocentní. S ohledem na popsané skutečností nemůže být v současnosti náhradou NetFlow a je třeba řešit problém spolehlivosti a bezpečnosti NetFlow dat alternativně.

Kapitola 3

sFlow

Alternativní možností pro monitorování provozu na síti je sFlow [16]. Tento protokol využívá odlišného principu pro sdílení dat než NetFlow, avšak problémy, s bezpečností a spolehlivostí jsou obdobné. Na rozdíl od NetFlow se tento standard o bezpečnosti a spolehlivosti zmiňuje a proto je zde popsán.

Za tímto standardem stojí společnost Hewlett-Packard. sFlow si klade za cíl umožnit monitorování vysokorychlostních sítí. Přístup, který realizuje sFlow, je stejně jako NetFlow založen na spolupráci agenta a kolektoru, kdy agent zasílá kolektoru zprávy v UDP paketu.

3.1 Princip

Hlavní rozdíl mezi nimi je ve zodpovědnosti za skládání paketů do toků. Jak je uvedeno v kapitole 1, v případě NetFlow je tato zodpovědnost delegována na agenta. Ten je vybaven pamětí, ve které si toky udržuje a po jejich ukončení informace o nich zasílá na kolektor. V případě sFlow je situace opačná. Agent pouze zajišťuje export vzorku paketů na kolektor, který tyto informace zpracovává.

V případě sFlow se počítá s tím, že agent bude vždy realizován pouze softwarovou formou na aktivním prvku (nebudou se užívat samostatné sondy). Z výše popsaného vyplývá skutečnost, že ne každý paket bude analyzován, ale předpokládá se pouze provádění vzorkování, tedy výběr každého n-tého paketu.

Zajímavou vlastností sFlow je konfigurace přes MIB, která poskytuje standardní mechanismus pro vzdálenou konfiguraci protokolem SNMP [16].

3.2 Přenášená data

Formát zprávy, kterou agent zasílá kolektoru, zobrazuje tabulka 3.1. Délky jednotlivých částí jsou proměnlivé v závislosti na typu předávaných dat (verze IP protokolu) nebo hodnotě předávaných dat (součástí bývají textové řetězce, např. URL).

Hlavička paketu je zkopírována z paketu, který byl předán vyšší vrstvě ke zpracování.

Rozhraní identifikuje rozhraní, kterým byl paket přijat i rozhraní, kterým byl dále odeslán.

Vzorkovací parametry nesou informaci o poměru, jak se vzorek vybírá a o počtu paketu, které mohly být vzorkovány (pakety přeskočené + počet vzorků).

hlavička paketu
rozhraní
vzorkovací parametry
další parametry paketu
identifikace uživatele
URL
čítače rozhraní

Tabulka 3.1: Formát sFlow zprávy

Další parametry paketu *(volitelné)* se ve standardu označují *forwarding*. Mohou obsahovat *next hop* pro daný paket, informace o zapouzdření v rámci VLAN, síťové masky zdrojové a cílové adresy, QoS, apod.

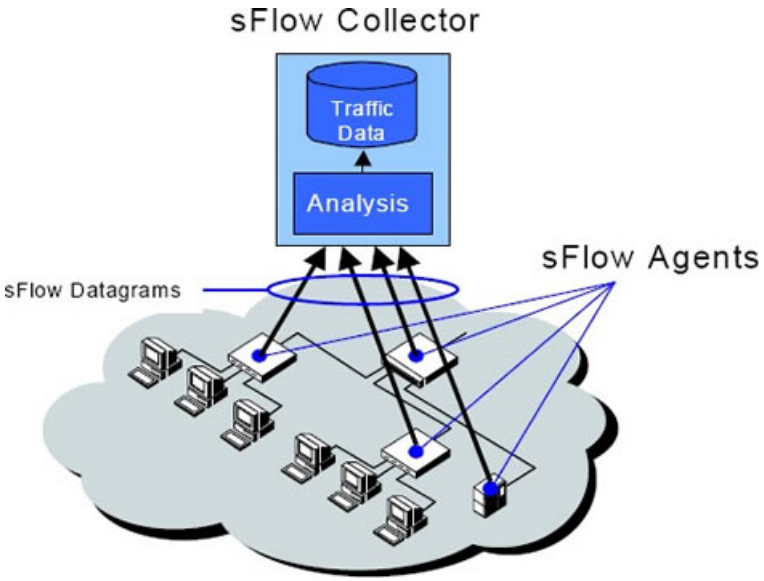
Identifikace uživatele *(volitelné)* se uplatní, pokud se uživatel před připojením do sítě musel autentizovat (např. pomocí služby RADIUS).

URL *(volitelné)* nese informaci o URL asociované s tokem a identifikuje, zda je asociovaná se zdrojem, nebo cílem.

Čítače rozhraní obsahují hodnoty, které jsou s paketem spojeny, tedy vstupní a výstupní rozhraní paketu.

V sFlow stejně jako v NetFlow není zabezpečení paketu definováno. Na rozdíl od NetFlow je tento problém ve standardu definován a jeho řešení je založeno na doporučení použít oddělené sítě pro správu, přes kterou budou tyto zprávy na kolektor doručovány [16]. Otázka spolehlivosti doručení není v tomto standardu řešena.

Vzorky jsou zasílány prostřednictvím UDP paketu na definovanou adresu a port. Nedostatečná spolehlivost transportního protokolu UDP není významná pro přesnost vzorků obdržených z sFlow agenta [16].



Obrázek 3.1: Architektura sFlow [4]

3.3 Shrnutí

sFlow může být náhradou NetFlow v případě, že cílem je sběr statistik o provozu sítě. V případě, že cílem je nasazení z důvodu účtování, pak sFlow není konkurentem NetFlow. Pokud má monitorování sloužit pro analýzu bezpečnosti, je konkurenceschopnost obou protokolů diskutabilní. V žádném případě nemůže být sFlow náhradou NetFlow z důvodu bezpečného přenosu zpráv o tocích.

Kapitola 4

Slabá místa monitorovacích systémů

Kapitola 4 se zaměřuje na analýzu tzv. „Signle Point of Failure" (SPoF) v monitorovacích systémech. SPoF je obecně jakýkoliv prostředek, jehož výpadek má za následek nedostupnost poskytované služby. Zmíněným prostředkem, může být aktivní prvek, linka spojující aktivní prvky, zdroj energie, server, apod. Eliminace těchto bodů je kritická především u služeb s vysokou dostupností (High Availability – HA). U takových služeb se obvykle udává procentuální dostupnost v roce (např. 99.9999 % odpovídá službě, která je nedostupná maximálně 31,5 sekund v roce). Aby bylo možné dosáhnout takovéto dostupnosti, používá se redundantní zajištění všech kritických zdrojů. Některým zdrojům nečiní problém existence alternativních zdrojů poskytujících stejnou službu (například WWW služby – více serverů může nabízet tentýž obsah). Na druhou stranu existují zdroje, které vyžadují existenci jediného prvku (například výchozí brána pro směrování – v systému není možné mít nastaveno více výchozích bran). U těchto zdrojů je nutno použít speciální zařízení, která se s tímto problémem dokáží vyrovnat.

V souvislosti s HA systémy, je nutno analyzovat dva případy:

- monitorování v HA síti a
- zajištění HA pro monitorovací systémy.

4.1 Monitorování sítí s vysokou dostupností

Je-li cílem monitorovat síť s vysokou dostupností je nutno počítat s tím, že se v takové síti vyskytuji redundantní prvky. Při monitorování provozu sítě se rozlišují agenti podle toho, zda získávají data

- z aktivního prvku pomocí replikace portů (příp. jsou jeho součástí) nebo
- zachytávají provoz na lince mezi aktivními prvky.

Pokud jsou agenti součástí aktivního prvku, pak je pouze potřeba sbírat data od všech agentů. Je-li místo agenta na aktivním prvku použitá sonda, pak nastává problém s omezeným počtem monitorovaných portů, kterými sonda disponuje. Je tedy nutné pořizovat další sondy. Výhodou tohoto zapojení s agentem na aktivním prvku nebo sondou je skutečnost, že výpadkem monitorovacího procesu (exportéru) nedochází k ovlivnění dostupnosti celé sítě.

Je-li monitorování realizováno pomocí odposlechu na lince (sonda je připojena pomocí Tap) pak je rozdíl, zda se jedná o agregovanou linku (viz dále) nebo o jednoduchou linku

mezi dvěma aktivními prvky. U jednoduché linky, kde se jedná pouze o redundantní aktivní prvky stačí zapojit na každou linku mezi dvěma zařízeními Tap a připojit sondu. Problém je stejný, jako v předchozím případě roste potřebný počet monitorovaných portů. Pokud se linka monitoruje pomoci Tap, pak neovlivňuje HA sítě, neboť Tap nemá vliv na linku (viz kapitola 1.1.

4.1.1 Agregované linky

Agregace linek je jedním ze způsobu, jakým lze zajistit redundanci linky a zároveň zvýšit její propustnost. Agregaci upravuje standard IEEE 802.3ad. Častěji se uplatňuje pouze pro zvýšení propustnosti sítě, neboť odolnosti vůči výpadkům efektivně chrání pouze před výpadkem rozhraní. Při výpadku zařízení se tato redundance neprojeví. Agregace může teoreticky poskytovat ochranu před narušením linky. Problémem však je, že se používá více kabelů v jednom svazku a typickým narušením je poškození celého svazku a tedy přerušení linky.

Problém monitorování agregovaných linek spočívá v tom, že na většině síťových zařízeních nelze replikovat agregované porty. Tento problém lze řešit použitím specializovaného Tap zařízení, které replikuje každou linku samostatně a je tedy nezbytné mít počet monitorovaných portů stejný jako počet linek v agregované skupině.

4.2 Zajištění HA pro monitorovací systémy

Jak bylo popsáno výše, eliminace SPoF se zajišťuje zvýšením redundance. V této situaci se ovšem problém rozpadá na dvě části redundance exportéru a redundance kolektoru. U exportéru je navíc nutné rozlišit, zda se jedná o agenta na aktivním prvku, nebo samostatnou sondu. V případě agenta není redundance nezbytná, neboť se jedná o součást aktivního prvku. Pokud selže celý aktivní prvek, pak není co monitorovat. U externích sond je situace odlišná, neboť dojde-li k jejich výpadku, pak se začne přicházet o data. V tomto případě je nezbytná redundance na úrovni sond, které budou zajišťovat sběr duplicitních dat. Zabezpečit spolehlivé doručení na kolektor není triviálním problémem a věnuje se mu další část této práce.

Duplicitní data

Redundancí exportéru vznikají duplicitní data, neboť více exportéru poskytuje informace o stejných tocích. Tento problém může být řešen dvěma přístupy. Jedním z přístupu může být implementace logiky, která dokáže označit dva toky za identické a vyloučit tedy redundanci informace. Druhým jednodušším přístupem je oddělení úložišť pro redundantní prvky a pracovat pouze s jedním úložištěm. Druhé úložiště využít pouze v případě, že v primárním úložišti nejdou data kompletní v důsledku výpadku exportéru pro primární úložiště.

Kapitola 5

Srovnání monitorovacích protokolu

V předchozích kapitolách byly popsány tři protokoly, které mohou být využity k monitorování provozu sítě. Důvody proč se provoz na síti monitoruje jsou různé a každý z těchto protokolů nabízí odlišný přístup pro jeho realizaci. Společně s tímto přístupem i odlišné vlastnosti. Přehled vlastností nabízí tabulka 5.1.

Vlastnost	NetFlow	IPFIX	sFlow
Standardizováno	RFC 3954	Několik standardů, některé stále nedokončeny	RFC 3176
Transportní protokol	UDP, SCTP*	SCTP, UDP*, TCP*	UDP
Podpora šifrování		DTLS, TLS*	
Podpora v hardware	√		√
Způsob sběru dat	Z paketu skládá toky a ty zasílá na kolektor		Sbírá vzorky paketů a jejich hlavičky zasílá na kolektor

* volitelné

Tabulka 5.1: Srovnání protokolů z hlediska cílů pro nasazení

V tabulce 5.1 je vidět, že podpora protokolu v hardware je v současné době rozšířená pouze u protokolu NetFlow a sFlow. To je ovlivněno především tím, že za těmito dvěma protokoly stojí přední výrobci síťových zařízení. Pokud jde o sFlow, pak agenti se nacházejí výhradně v aktivních prvcích a je tedy nezbytné, aby jej aktivní prvky podporovaly. U NetFlow je situace poněkud odlišnější. Podpora tohoto protokolu není výhradně na aktivních prvcích, ale jsou dostupná samostatná zařízení zajišťující získávání informací o provozu sítě. Díky tomu je možné nasadit NetFlow i v sítí bez aktivních prvků podporujících tento protokol.

Ze srovnání protokolů z hlediska podpory spolehlivého a důvěrného přenosu vychází jako jednoznačný vítěz protokol IPFIX, který se na rozdíl od svých konkurentů zabývá i bezpečností. Pokud tedy neexistuje důvod nasadit pro monitorováni provozu sFlow, pak srovnání NetFlow a IPFIX jednoznačně nahrává protokolu IPFIX. Jak již bylo uvedeno v kapitole 2.3, NetFlow stále poráží IPFIX díky svému rozšíření a podpoře v hardware. Vzhledem k tomu, že situace kolem IPFIX, ačkoliv je na dobré cestě, není stále zcela vyjasněna, nelze vyloučit i možnost, že se nikdy nedočká rozšíření jako NetFlow. Protože povinnost monitorovat provoz na síti je v mnoha zemích světa vyžadována zákonem, nelze se jí vyhnout. Pokud už tedy NetFlow musí být nasazen, je žádoucí, aby bylo uděláno maximum pro zabezpečení těchto dat. V dalších kapitolách se tato práce zabývá možnostmi zabezpečení NetFlow dat z hlediska spolehlivosti a důvěrnosti přenosu. Je nutno ještě poznamenat, že cílem je zajistit důvěrný a spolehlivý přenos mezi dvěma či více vzdálenými lokalitami, které jsou propojeny

skrze veřejnou síť.

Bezpečnostní cíl	NetFlow	IPFIX	sFlow
Důvěrnost		DTLS*: SCTP*, UDP TLS: TCP	
Integrita		DTLS*: SCTP*, UDP TLS: TCP	
Autentizace		DTLS*: SCTP*, UDP TLS: TCP	

* povinně podporované protokoly

Tabulka 5.2: Srovnání protokolů z hlediska bezpečnosti

Kapitola 6

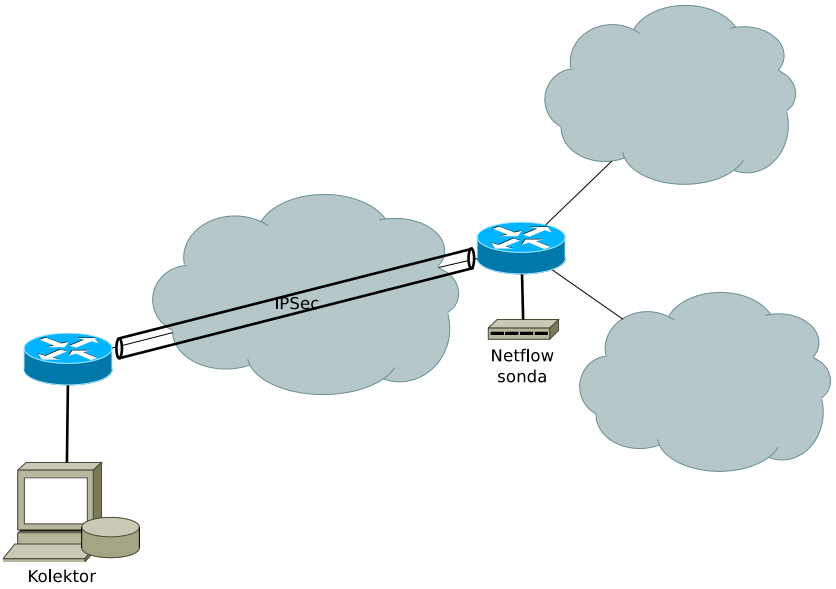
Návrh řešení

Tato kapitola se zabývá zkoumáním možných řešení pro zabezpečení transportu dat z exportéru na kolektor. Cílem tohoto zabezpečení je zajistit spolehlivé doručení na kolektor a utajení dat. Standart popisující NetFlow tuto problematiku nezmiňuje, proto byly jako výchozí bod použita doporučení ze standardu k protokolu sFlow (viz kapitola 3).

6.1 Zabezpečená privátní síť

Možnost, která je zmíněna v standardu sFlow je využití privátní sítě. Privátní síť může být vytvořena pomocí fyzického nebo logického oddělení od sítě veřejné. Protože cílem je zajistit transport dat skrze veřejnou infrastrukturu, možnost fyzického oddělení nepřichází v úvahu. Logické oddělení je možné realizovat na linkové (VLAN) nebo síťové vrstvě (VPN). Protože přístup k veřejné síti je omezen na třetí vrstvu, nelze použít oddělení pomocí VLAN.

Zbývající variantou zůstává virtuální privátní síť. Nejpoužívanější technologií pro realizaci VPN je v současné době IPSec [13, 21]. Schéma zapojení realizované tímto způsobem je zobrazeno na obrázku 6.1.



Obrázek 6.1: VPN tunel

6.1.1 Vlastnosti

Běžně se rozlišují dva typy VPN:

- Síť - Síť,
- Host - Síť (přístupová VPN).

Přístupová VPN se obvykle používá pro přístup mobilního uživatele ke zdrojům v privátní síti. V tomto případě uživatel musí spojení inicializovat a na jeho počítači se vytvoří virtuální síťové rozhraní, které slouží pro komunikaci. V druhém případě je tunel pro uživatele skryt. Paket, který nese data určená do vzdálené sítě, se celý zabalí do jiného paketu na hraničním prvku sítě a pošle přes veřejnou síť k zařízení, které představuje druhý konec tunelu. Tam se rozbalí a dále pokračuje původní paket. Pro zabalení paketu a odeslání přes veřejnou síť se používají protokoly GRE (Generic Routing Encapsulation) [8], MPLS (Multiprotocol Label Switching) [19], IPSec.

V případě GRE se původní paket opatří hlavičkou GRE protokolu a vloží se jako datová část síťového protokolu, kterým bude doručen k cíli. Výhodou tohoto protokolu je kompatibilita s různými síťovými protokoly.

MPLS stejně jako GRE může být přenášen v různých protokolech. Základní rozdíl spočívá v převodu IP adresy, paketu jenž zapouzdřuje na „popisek", pomocí kterého se hledá cesta k cíli. Toto chování emuluje přepínání okruhů známe ze sítí ATM nebo Frame Relay. Sám o sobě tento protokol nezajišťuje utajení. K tomu se užívá v kombinaci s IPSec.

IPSec dovoluje dva režimy:

- transportní a
- tunelový.

V případě transportního režimu jde o zabezpečení od transportní vrstvy výše. V tomto případě vyžadují oba dva komunikující body veřejnou IP adresu a podporu IPSec. V tunelovém režimu pracuje stejně jako tunel GRE, tedy pro komunikující uzly je transparentní. IPSec nabízí odděleně zajištění autenticity (AH – Autentication Header) a utajení (ESP – Encapsulating Security Payload).

Pokud se vhodně zvolí způsob realizace VPN pak její konfigurace je pro exportér transparentní a nevyžaduje jeho rekonfiguraci. Problém může být s nastavením VPN pokud se propojuje více lokalit, neboť narůstá složitost správy.

6.1.2 Splnění požadavků

Z popisu použitých technologií vyplývá, že ne všechny mohou splnit kritéria pro zabezpečení přenosu protokolu NetFlow.

Utajení

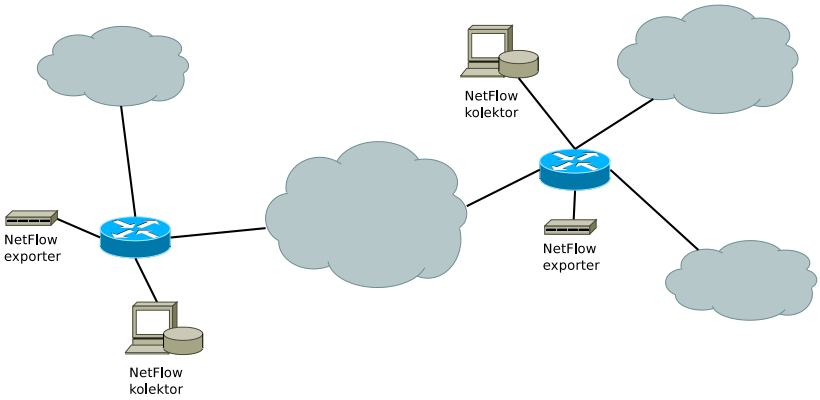
V tomto případě je cílem zajištění důvěrnost dat a proto je tedy nutné uvažovat pouze o tunelech šifrovaných. Nasazení této technologie přispěje k zajištění důvěrnost dat.

Spolehlivost

Ze samotné definice vyplývá, že se jedná o síť virtuální, která je realizovaná nad veřejnou sítí. Tato definice ovšem nesplňuje všechna kritéria, která byla na začátku stanovena. Neřeší problém spolehlivosti přenosu a ze samotného principu spolehlivost garantovat nemůže, neboť rozpadne-li se spoj na veřejné sítí, je zrušen i tunel mezi dvěma konci privátní sítě a všechny odeslané pakety nesoucí UDP data jsou zahazovány. Řešení, které tedy bude implementováno, musí mít možnost zachytit všechny pakety bez ohledu na stav veřejné sítě.

6.2 Lokální sběr dat

Předchozí řešení umožnilo pomoci virtuální privátní sítě zajistit důvěrný přenos dat přes veřejnou síť. Ovšem ukázalo se, že není možné pomocí této technologie zabránit ztrátě dat a zajistit tedy jejich spolehlivé doručení. Problémem v tomto případě byla nutnost spoléhat na stabilitu a spolehlivost linky poskytovatele připojení. Je-li cílem zajistit spolehlivý sběr dat, pak je nezbytné zajistit, aby byl kolektor propojen s exportérem přímou linkou fyzicky oddělenou od běžného provozu nebo při použití sdílených prostředků oddělením pomocí VLAN. Zapojení je vidět na obrázku 6.2.



Obrázek 6.2: Lokální umístění kolektorů

6.2.1 Vlastnosti

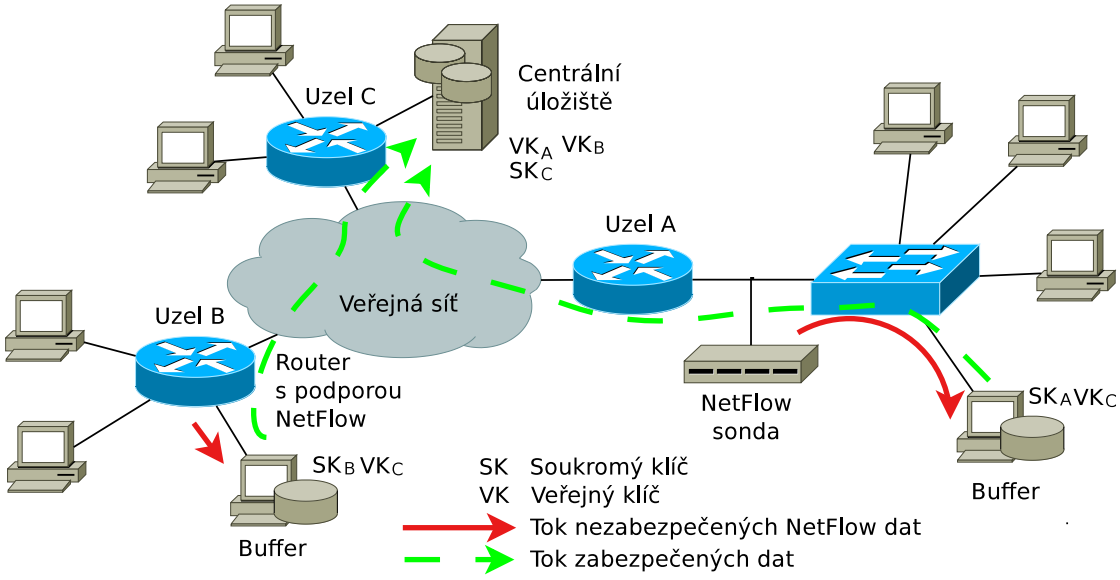
Oddělení sítě pomocí VLAN, resp. fyzické, které není směrováno, resp. připojeno do veřejné sítě umožní zasílat data v otevřené formě, protože není možnost tato data odposlechnout. Použití VLAN sebou ovšem přináší riziko přetížení sdílených prostředků a tedy možnost ztráty dat.

6.2.2 Splnění požadavků

Zapojení poskytuje oddělenou privátní síť pro monitorovaný provoz a z tohoto pohledu v dané lokalitě zajišťují minimálně stejné vlastnosti jako předchozí řešení.

Utajení

V tomto případě nedochází k šifrování dat, ale utajení je zajištěno jejich nepřístupností. Tento cíl je tedy splněn.



Obrázek 6.3: Zapojení s lokálním kolektorem

Spolehlivost

Použití nespolehlivého transportního protokolu a nemožnost opětovně požádat o přenos dat vždy představuje riziko jejich ztráty. Ideálním řešením by byla záměna transportního protokolu. Taková změna je ovšem nemožná, neboť ji nelze aplikovat na libovolné NetFlow zařízení. Protože takovéto řešení nepřichází v úvahu je tento koncept nejvhodnějším řešením pro zajištění spolehlivého sběru dat.

6.2.3 Další vlastnosti

Ačkoliv toto řešení nabízí utajení i spolehlivost, ignoruje požadavek na přenos mezi lokalitami skrze veřejnou síť a nesplňuje tedy všechny požadavky.

6.3 Kombinace předchozích možností

Z rozboru předchozích řešení vyplývá, že problémem není zajistit utajení exportovaných dat, ale garantovat jejich spolehlivé doručení. Nejvyšší míru spolehlivosti nabízí řešení, kdy je kolektor umístěn ve stejné lokalitě a připojen pomocí dedikované linky a je tedy minimalizován vliv provozních dat na spolehlivost přenosu monitorovacích dat. Cílem je ovšem doručení na centrální kolektor. Jsou-li již data spolehlivě doručena na lokální kolektor, zbývá zajištění jejich důvěrného a spolehlivého transportu na centrální kolektor. Takovéto doručení se zajistí použitím spolehlivého transportního protokolu v kombinaci s protokolem zajišťujícím důvěrnost dat.

Na obrázku 6.3 je zobrazen příklad zapojení, který využívá lokální kolektory (označeny „Buffer") zajišťující spolehlivý sběr NetFlow dat. Tato zařízení autonomně dopravují data na centrální kolektor pomocí spolehlivého a bezpečného protokolu. V navrhovaném schématu je užito asymetrické kryptografie pro autentizaci komunikujících stran. Spojení inicializuje „buffer" má-li k dispozici data pro doručení. Po připojení ke kolektoru se autentizuje pomocí svého soukromého klíče. Aby se ověřilo, že data zasílá na správný kolektor, autentizuje se

kolektor svým soukromým klíčem „bufferu". Po ukončení přenosu se spojení přeruší a čeká se dokud nejsou k dispozici další data. Tok nezabezpečených dat představuje přenos NetFlow v otevřené podobě pomocí nespolehlivého protokolu. Data jsou oddělena od zbytku sítě pomocí VLAN, je tedy zajištěna jejich důvěrnost. Zabezpečený tok představuje přenos dat s garantovaným doručením a utajením.

6.3.1 Vlastnosti

Tento způsob řešení zadaného problému zajištění zabezpečeného transportu monitorovacích systému nabízí vypořádání se všemi problémy, které byly během analýzy protokolu objeveny. Úspěch je vykoupen požadavkem na umístění lokálního kolektoru zajišťující buffer v každé oblasti, kde se nacházejí sondy.

6.3.2 Splnění požadavků

Toto zapojení kombinuje obě předchozí řešení, kdy jedno řešení nebylo schopno zajistit spolehlivost a druhé doručení na vzdáleny kolektor.

Utajení

Data jsou ze sondy získána pomocí oddělené sítě, tedy bezpečně doručena na buffer a z toho jsou přeposlána na vzdálený kolektor, například skrze zašifrovaný VPN tunel. Data tedy nikde po veřejné síti nejdou v otevřené formě a jejich utajení je zajištěno.

Spolehlivost

Doručení zprávy z exportéru na buffer probíhá stále nespolehlivě, ale pravděpodobnost výpadku je v této situaci minimální. Z bufferu na kolektor již může být použit transportní protokol garantující spolehlivost. V tomto případě lze hovořit i o splnění cíle spolehlivého doručení.

6.3.3 Možnost implementace

Nyní budou rozebrány možnosti, jak implementovat buffer, který bude přijímat zprávy od agentů a dále je přeposílat na kolektor. Přeposílání musí poskytovat všechny požadované vlastnosti. Zajistit spolehlivý přenos je v případě možnosti ovlivnit chování vysílací i přijímací strany triviálním problémem. Vhodná volba transportního protokolu, který zajistí spolehlivé doručení, řeší celý problém. Takovým protokolem může být protokol TCP nebo SCTP, který se velmi často objevuje v návrzích nových protokolů.

Důvěrnost by mohlo zajistit zbudování bezpečného tunelu mezi bufferem a kolektorem pomocí VPN. Avšak tato možnost není optimální, neboť v případě více než dvou lokalit vede na složitou zprávu. Ideálním způsobem jak zajistit bezpečný přenos, je vytvoření dvoubodového spojení přímo při komunikaci. Tuto vlastnost nabízí například protokol TLS. Dalším protokolem, který poskytuje šifrované spojení, je protokol SSH.

Existující nástroje

Součástí analýzy protokolu NetFlow byl i průzkum existujících nástrojů. Z průzkumu bylo zjištěno, že žádný dostupný software pro NetFlow nezajišťuje způsob, jak garantovat spolehlivé doručení na vzdálený kolektor. Zároveň z diskuze s konzultantem z CVIS VUT, který

Obrázek 6.4: Příklad softwarové sondy FlowMon Probe 2000 [23]

zadával požadavek na řešení tohoto problému, vyplynulo, že nejrozšířenější aplikací pro práci s NetFlow daty je balík aplikací *nfdump* [10]. To potvrzuje i skutečnost, že tento balík bývá součástí běžně dodávaných softwarových nebo hardwarově akcelerovaných sond, například FlowMon Probe 2000 od společnosti Invea-Tech a.s. (obr. 6.4).

Balík aplikací *nfdump* se skládá ze čtyř aplikací:

nfcapd představuje aplikaci poskytující služby kolektoru, naslouchá tedy na stanoveném portu a ukládá přijatá data do souboru.

nfdump slouží pro zpracování souborů s daty vytvořených aplikací *nfcapd*. Nad tímto souborem provádí analýzy, filtraci a zobrazuje výsledky.

nfprofile zpracovává soubory vytvořené aplikací *nfcapd* a rozčleňuje data do „profilů" podle stanovených kritérií.

nfreplay umožní přehrát soubory získané pomocí aplikace *nfcapd* zpátky do sítě. Přehrávaná data mohou být omezena filtrem.

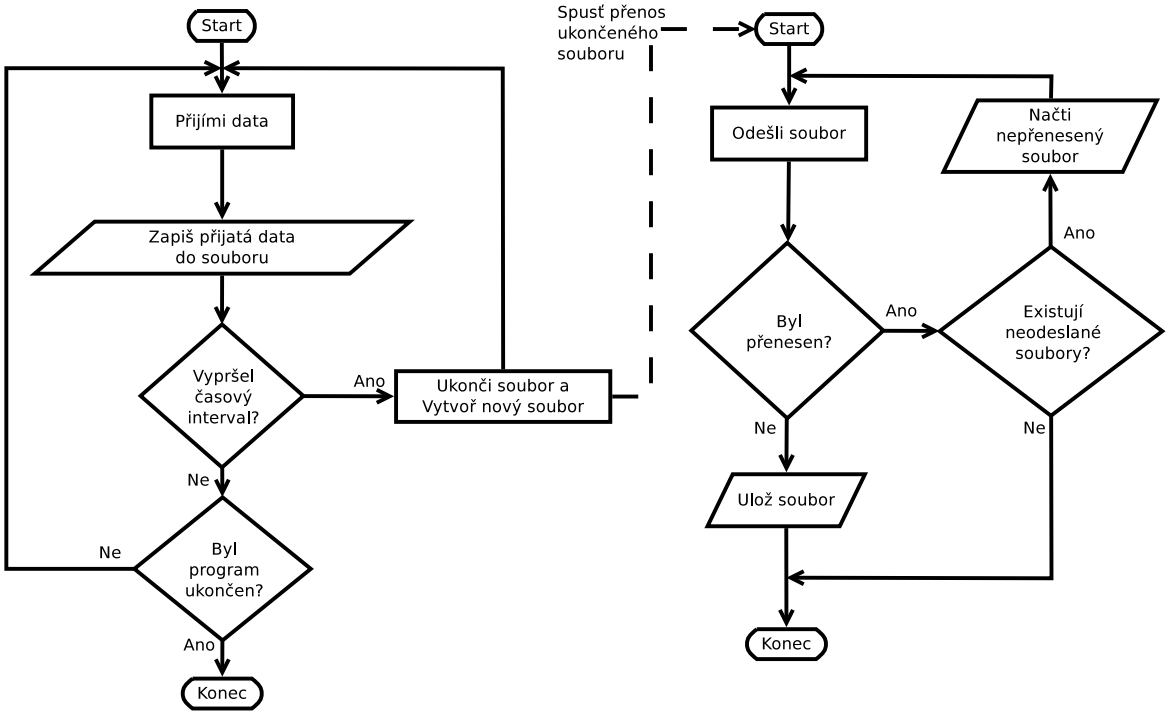
Návrh algoritmu bufferu

Na základě shrnutí požadavků byl navržen algoritmus použití lokálního bufferu, který lze popsat diagramem na obrázku 6.5.

Patří-li balík aplikací *nfdump* mezi nejrozšířenější, nabízí se jako vhodné řešení využít jeho vlastností jako základu pro buffer. Tímto se splní první část algoritmu příjmu dat. Zároveň tato aplikace splňuje i bod 2, tedy začít psát do nového souboru po vypršení času. Je tedy nutné zajistit aby se po rotaci soubor přenesl na cílový stroj. Tuto činnost již aplikace zajistit neumí, ale její součástí je možnost zavolání externího příkazu. Protože tato aplikace je dostupna pod licencí *open source* a umožňuje po přetočení souboru zavolat externí aplikaci. Mělo by být jednoduché upravit aplikaci tak, aby nevolala externí program, ale aby zajistila přenos souboru na cílový kolektor.

Transport souborů

Volba vhodného transportního protokolu může implementaci algoritmu výrazně zjednodušit. Výhodou jednodušší implementace je nižší riziko chyby. Jak bylo uvedeno výše, pro



Obrázek 6.5: Diagram algoritmu pro zajištěni doručení dat na centrální kolektor

zabezpečený transport souborů se nabízejí dvě možnosti:

TLS je běžně rozšířenou knihovnou pro realizaci zabezpečeného přenosu. Tuto variantu využívají běžně protokoly HTTP, LDAP, aj. Výhodou tohoto protokolu je podpora asymetrické kryptografie založené na certifikátech podle standardu X.509.

SSH je jiný způsob implementace šifrovaného spojení. Stejně jako TLS je založen na asymetrické kryptografii, ale na rozdíl od TLS používá vlastní formát, který není kompatibilní se standardem X.509.

Transport souborů vyžaduje podporu na bufferu i na kolektoru, kam jsou soubory přenášeny. Aby nebylo nutné implementovat serverovou část (na kolektoru), která budou pouze přijímat a ukládat data. Nabízí se možnost použití existujícího protokolu pro přenos souborů.

Možnosti využití standardu X.509

Standard X.509 [12, 7] je nejrozšířenějším způsobem distribuce a správy veřejných klíčů používaných v asymetrické kryptografii. Certifikát veřejného klíče je datová struktura skládající se z části datové a podpisové. Datová část obsahuje otevřený text identifikující entitu. Minimálním obsahem musí být veřejný klíč a textový řetězec popisující entitu. Podpisová část obsahuje digitální podpis certifikační autority datové části [15].

Možnost využití certifikátu se nabízí při nasazení transportního protokolu TLS. S použitím tohoto standardu je spojená potřeba existence certifikační autority, která tyto certifikáty podepisuje. Tento požadavek může být vyřešen podepsáním certifikátu vlastní autoritou, avšak tento způsob narušuje celý význam standardu X.509.

Volba výchozího protokolu

S přihlédnutím k nutnosti zajišťovat podpis veřejného klíče při použití transportního protokolu TLS, byl zvolen výchozím transportním protokolem protokol SSH. Důvodem pro tento krok byla skutečnost, že na počítačích s operačním systémem Linux je jeho podpora implicitní a součástí distribucí bývá obvykle i SSH server, což se o jiných aplikacích podporujících zabezpečený přenos souborů nedá říci. Výsledkem této volby jsou minimální požadavky na konfiguraci kolektoru. Implementace klienta je v tomto ohledu založena pouze na využití knihovních funkcí.

Ošetření opětovného přenosu při neúspěšném spojení

Pokud na lince dochází ke ztrátovosti paketů, pak jejich opětovné přenesení zajistí protokol TCP autonomně a není třeba tento problém řešit. Problém nastává, když se spojení úplně rozpadne, nebo se ho nepodaří vůbec navázat a přenos se tedy vůbec neuskuteční. V takovém případě musí být opětovný přenos zajištěn aplikací. Implementace tohoto chování není problémem, protože knihovní funkce využité pro přenos skončí s chybou, není-li dokončen úspěšně a je tedy možné patřičně zareagovat.

Implementace

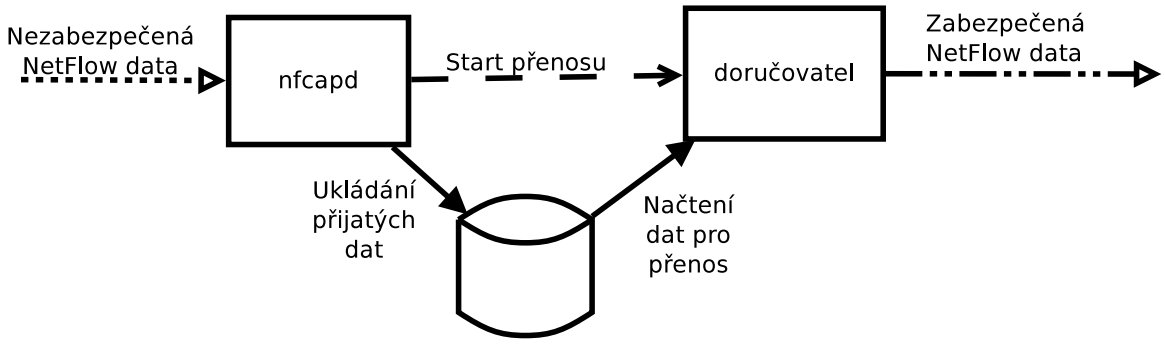
Předtím, než byla zahájena implementace, byla provedena analýza možností nasazení upravené aplikace na sondě FlowMon Probe přístupné v laboratoři. Z dokumentace bylo zjištěno, že sonda je provozována na Linuxu distribuce CentOS, ke které má uživatel přístup formou uživatelského účtu `flowmon`, avšak nemá přístup s právy správce. Přístup, který je k dispozici, dovolí konfigurovat parametry sondy pro sběr dat. Umožní i zkopírovat nové soubory na sondu do domácího adresáře, ale v žádném případě není možné vyměnit existující aplikace, které zajišťují sběr dat. Tento problém by bylo možné obejít umístěním aplikace do domácího adresáře a úpravou spouštěcích skriptů. Tuto variantu není možné uplatnit v navrhované podobě, neboť dostupná oprávnění vylučují editaci těchto skriptů. Možnosti pro realizaci této myšlenky je konfigurace vlastního startovacího skriptu a zajistit jeho spuštění jiným způsobem.

Alternativní řešení

Jinou možností může být využití vlastností, které nabízí aplikace, která již na sondě k dispozici je. Dříve bylo uvedeno, že aplikace disponuje možností zavolat externí aplikaci po přetočení souboru, do kterého byl ukončen zápis. Nabízí se možnost implementovat samostatně aplikaci, která dostane název souboru, který má být přenesen a tento úkol zajistí, včetně situace, kdy se přenos nezdaří. Ověření, zda lze takto sondu nastavit, dopadlo úspěšně.

V tuto chvíli již není zapotřebí aplikace, která bude naslouchat příchozím paketům NetFlow protokolu. Místo toho stačí aplikace (dále nazývaná „doručovatel"), která na vstupu dostane cestu k souboru, který má být přenesen na vzdáleného hosta. V případě, že se tento přenos nezdaří si zapamatuje, že soubor nebyl přenesen. Pokud byl přenos úspěšný, zkontroluje, jestli neexistuje nějaký nepřenesený soubor a pokusit se ho přenést.

Pro tento úkol se nabízí možnosti vytvořit aplikaci nebo využít existujících aplikací a přenos zajistit pomocí skriptu. Splnění úkolu je možné dosáhnout využitím standardních aplikací operačního systému Linux. Tato skutečnost upřednostňuje realizaci pomocí skriptu před tvorbou binární aplikace, neboť zvyšuje míru přenositelnosti mezi různými distribucemi.



Obrázek 6.6: Schéma aplikace

6.4 Shrnutí

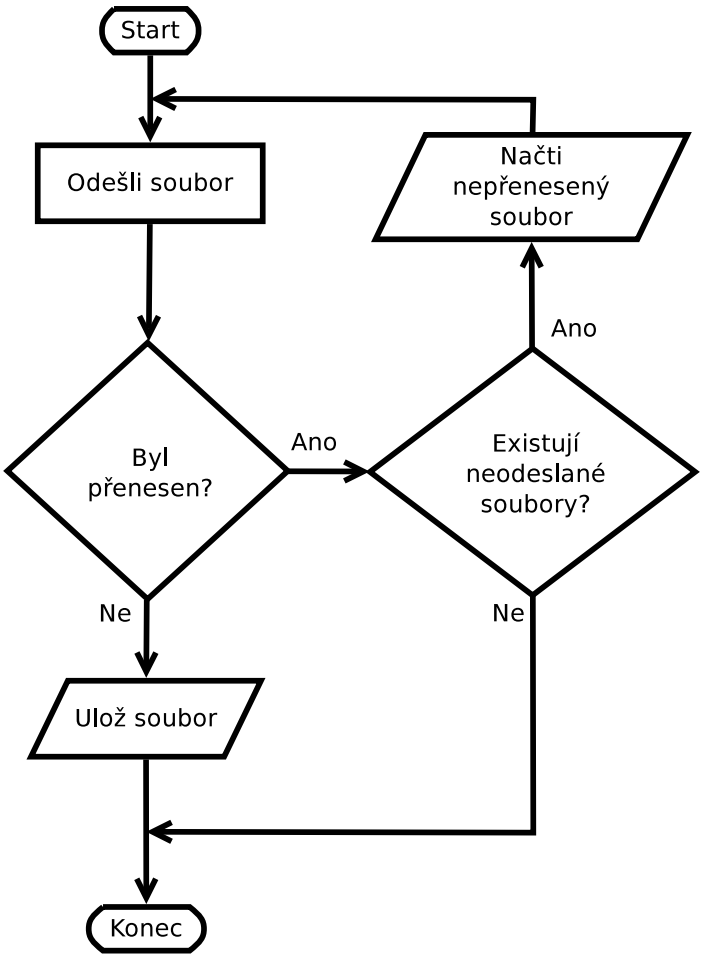
Po důkladné analýze možností, které se nabízely pro řešení požadovaného problému bylo zjištěno, že jediným možným řešením je nasazení lokálního bufferu, který jediný dokáže poskytnout nejvyšší odolnost proti ztrátě dat. Největší nevýhodu tohoto řešení lze vidět v požadavku na další stroj do každé lokality, kde se nachází alespoň jedna sonda. Po zjištění skutečnosti, že sondy dodávané na trh firmou Invea-Tech a.s. jsou založeny na operačním systému Linux se objevila myšlenka vytvořit buffer přímo na nich. Drobnou komplikaci činila skutečnost, že pro manipulaci se sondou jsou jen omezená práva, nicméně to nebylo překážkou pro realizaci této myšlenky.

Protože nemohla být provedena aktualizace samotné aplikace zajišťující sběr dat, bylo přistoupeno k realizaci formou externí aplikace. Po zvážení všech možností se jevilo jako nejlepší realizovat buffer pomocí standardních linuxových nástrojů ovládaných skriptem. V případě, že by se během vývoje ukázalo toto řešení jako nevyhovující, stále existuje možnost vrátit se k realizaci formou binární aplikace.

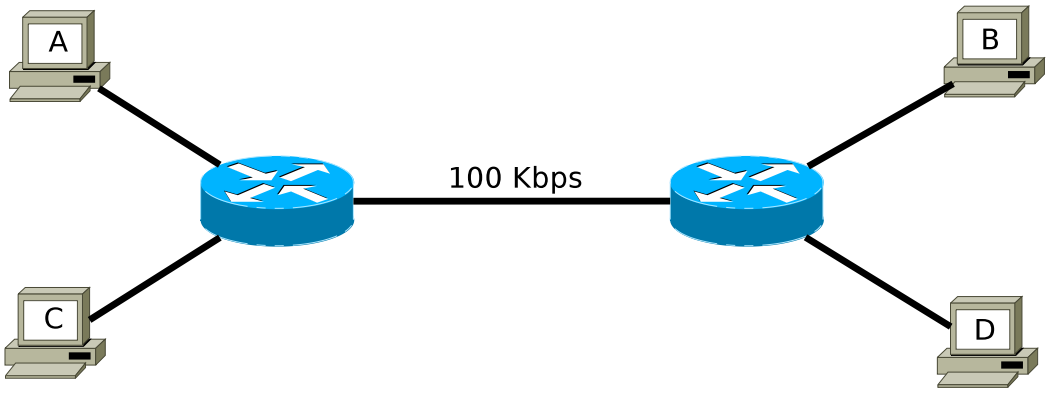
Kapitola 7

Realizace

Realizace „doručovatele" probíhala v mnoha iteracích proložených testováním a konzultacemi se správcem z CVIS VUT, neboť prioritní nasazení výsledků se uvažuje v síti VUT. Protože bylo přistoupeno k využití existující aplikace *nfcapd* pro příjem NetFlow dat bylo třeba realizovat „doručovatele" (schéma činnosti na obr. 7.1) a ověřit funkčnost při spojení s aplikací *nfcapd* a zapojení na sondě.



Obrázek 7.1: Algoritmus činnosti „doručovatele"



Obrázek 7.2: Schéma zapojení při testování přetížené linky

7.1 První verze

První navržená verze byla velice triviální, jejím cílem bylo ověřit, zda pomocí standardních nástrojů sestavit „doručovatele", který zajistí přenos souboru na vzdáleného hosta. V případě, že se přenos nezdaří se aplikace ukončí. O další přenos se pokusí automaticky při jejím dalším spuštění.

Realizace a test této verze byl prováděn mezi dvěma počítači, bez nutnosti konfigurace sondy. V předchozí kapitole bylo navrženo zabezpečení dat pomocí protokolu SSH. Pro tento účel byla tedy zvolena aplikace *scp*, která je standardní součástí operačního systému Linux. Pomocí dalších standardních nástrojů (cp, find, test, ...) bylo navrženo zjištění úspěchu doručení a zapamatování souboru v případě neúspěchu.

Výpadek linky byl simulován fyzickým rozpojením. Výsledkem testu této verze bylo očekávané chování. Tedy nebylo-li k dispozici spojení na cílového hosta, doručení bylo zajištěno při opětovném spuštění aplikace, kdy už bylo spojení aktivní. Pokud se nepodařil přenos více souboru po sobě, byly všechny úspěšně přeneseny při posledním úspěšném spojení.

7.2 Druhá verze – zabránit přetížení linky

Jako ochrana před posíláním dat při přetížení linky byl zvolen mechanismus zkrácení doby čekání pro navázání spojení. Prvním příznakem přetížené linky je obvykle právě prodloužení doby, kterou paket stráví ve frontě na aktivním prvku, případně ztrátovost paketů. Doplnění této možnosti nebylo implementačně náročné, mnohem náročnější bylo ověření, že tento algoritmus skutečně funguje, neboť vyžadoval složitější testování.

Tento test byl opět prováděn pomocí dvou počítačů. Schéma zapojení znázorňuje obrázek 7.2.

V tomto zapojení byla linka mezi směrovači nastavena na propustnost 100 Kb/s. Nejprve se provedl přenos souboru mezi počítači A a B bez dalšího provozu na této lince pro ověření zapojení. V druhé fázi byl generován provoz mezi počítači C a D, jehož cílem bylo přetížit linku. K tomuto účelu posloužil přenos velkých souborů. Při takto zatížené lince byl spuštěn „doručovatel".

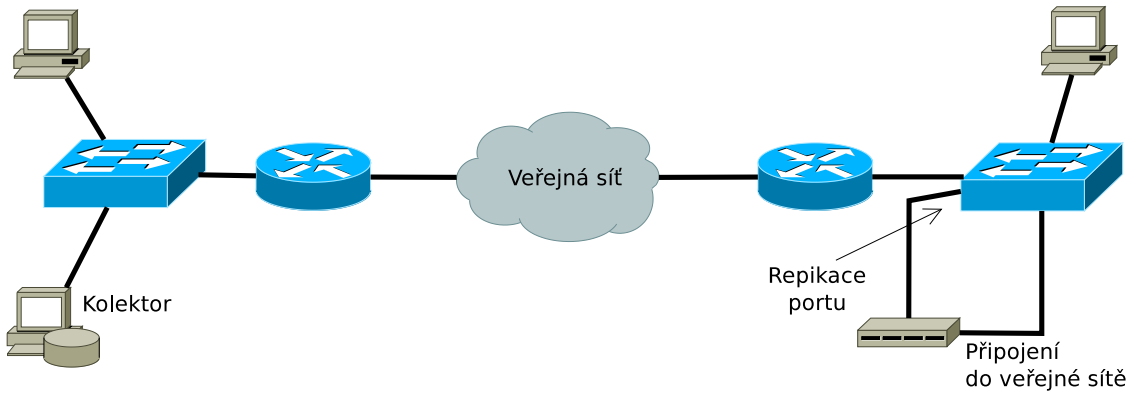
Protože doba čekání na odpověď byla uměle zkrácena na 1 sekundu (oproti standardním 30 sekundám) došlo k expiraci spojení dříve než bylo navázáno. Chování bylo stejné jako v případě, že linka není dostupná. K přenosu všech nepřenesených souboru došlo později, kdy již byla linka opět dostupná.

7.3 Hierarchie cílových souborů

Další požadavek na doplnění byl vyvolán z CVIS. Software obstarávající zachytávání souborů dovoluje přepínačem ukládat souboru hierarchicky podle data, kde cílový strom je ve formátu `rok/měsíc/den/`. Požadavkem bylo, aby tato struktura souboru byla zachována i na cílovém kolektoru. Implementace tohoto požadavku si vyžádala úpravy ohledně nutnosti explicitního vytvoření cílové struktury na kolektoru, neboť použitá aplikace nedokáže zajistit toto chování autonomně.

7.4 Testovací prostředí

Po ověření základních principů na zjednodušeném prostředí sestaveného pomocí počítačů bylo sestaveno testovací prostředí, které se podobá reálnému prostředí, ve kterém mají být výsledky uplatněny. Schéma prostředí je zobrazeno na obrázku 7.3. Použitou sondou je sonda Flowmon Probe 2000.



Obrázek 7.3: Schéma testovacího prostředí

Konfigurace v tomto případě spočívala ve vytvoření uživatelského účtu na kolektoru, pod kterým bude „doručovatel" přistupovat, aby tam přesunula nově vzniklé soubory. Dalším krokem bylo zajištění autentizace uživatele bez nutnosti zadávat heslo. To se zajistí pomocí vygenerování ssh-klíčů a zkopírování veřejného klíče na kolektor. Dále také musela být na sondu umístěna aplikace zajišťující činnost „doručovatele". Nastavení „doručovatele" se provádí volbou následujících parametrů (příklad konfigurace obr. 7.4):

- uživatelské jméno,
- soubor s privátním klíčem,
- adresa kolektoru,
- cílová složka,
- složka pro dočasné uložení neodeslaných souborů.

V poslední řadě se provede nastavení sondy, aby po přetočení logu zavolala aplikaci, která zajistí doručení na kolektor. Příklad nastavení je vidět na obrázku 7.5.

Po ověření činnosti zapojení bylo přistoupeno k analýze chování tohoto zapojení. Při testování v laboratorních podmínkách představuje problém nemožnost vygenerování stejného

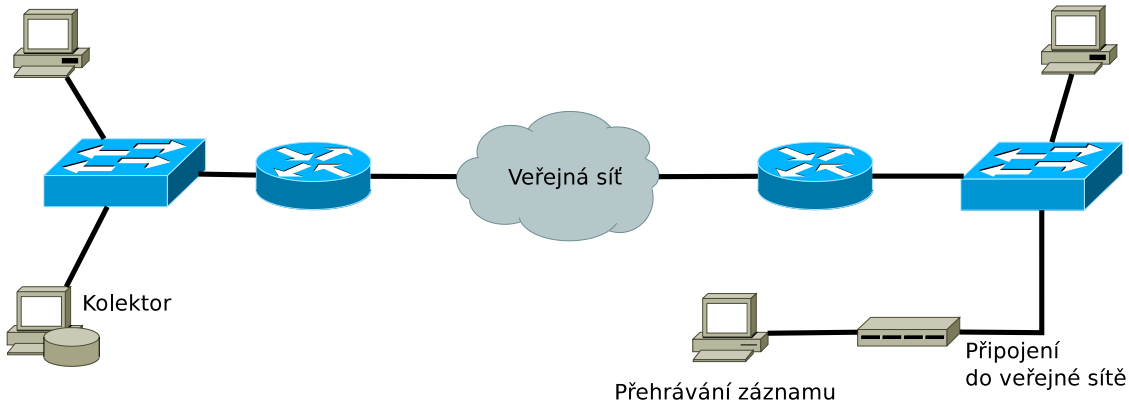
```
USER=kolektor
KEY=/home/flowmon/.ssh/id_rsa
SERVER=10.10.10.103
DEST_DIR=/lokalita_A
DIR_NOSEND='/tmp/nfcupd_nosend'
```

Obrázek 7.4: Příklad konfigurace „doručovatele"

```
%sources = (
'in'  => { 'port' => '3000', 'col' => '#000000', 'type' => 'netflow',
           'optarg' => '-x "/home/flowmon/deliver.sh %f in"' },
'out' => { 'port' => '3001', 'col' => '#000000', 'type' => 'netflow',
           'optarg' => '-x "/home/flowmon/deliver.sh %f out"' },
)
```

Obrázek 7.5: Příklad konfigurace sondy

počtu toků, jako v reálném prostředí. Aby bylo možné ověřit činnost navrženého řešení se zátěží odpovídající reálnému provozu, byla sonda využitá pouze jako buffer a data na ni byla zasílána z počítače pomocí aplikace *nfreplay* přehrávající anonymizovaný záznam z reálného provozu (upravené schéma znázorňuje obrázek 7.6). Provedená změna je pouze na straně kolektoru a nemá tedy vliv na výsledky testování. V důsledku této změny se velikost vzniklého logu, který bylo nutno přenést z bufferu na kolektor, pohybovala nejčastěji v rozmezí 50 - 100 MB. Přehledy hodnot přenosu, bez výpadku linky, jsou shrnuty v tabulce 7.1.



Obrázek 7.6: Schéma testovacího prostředí s upraveným kolektorem

V další fázi bylo přistoupeno k testování nestandardních situací:

- Dlouhodobý výpadek,
- Krátkodobý výpadek v době přenosu,
- Opakované výpadky linky na krátké okamžiky.

Pro všechny testy bylo využito existující konfigurace bufferu a kolektoru. A exportér (realizován pomocí *nfreplay*) zasílal v každém testu stejná data. Na konci testu se provádělo

Měřená vlastnost	Hodnota
Celková doba testování	6 hodin
Počet přenesených souborů	72
Interval přenosu souborů	5 minut
Rychlost linky	100 Mb/s
Nejmenší velikost přenášeného souboru	41 MB
Maximální velikost přenášeného souboru	110 MB
Průměrná velikost přenášeného souboru	82 MB
Medián velikosti přenášených souborů	93 MB
Nejkratší doba přenosu souboru	4,3 s
Nejdelší doba přenosu souboru	17,8 s
Průměrná doba přenosu souborů	9,6 s
Medián přenosu souborů	10,1 s

Tabulka 7.1: Charakteristiky přenosu

porovnávání obsahu přenesených dat. V případě, že test byl kratší, byla ověřována konzistence jen skutečně exportovaných dat.

7.4.1 Dlouhodobý výpadek

Cílem tohoto výpadku bylo ověření schopností vyrovnat se s větším množstvím nepřenesených souborů (popis viz tabulka 7.2). V tomto případě bylo zjištěno, že při velkém množství souborů, jejichž přenos překročí dobu periody pro přetočení souboru na bufferu, dojde k otevření dalšího spojení, kterým se nový soubor přenáší. Toto není samo o sobě problémem, naopak je to žádoucí, protože v případě, že je linka v pořádku je zajištěno, že nejnovější soubory jsou doručeny co nejdříve. Ovšem byl zjištěn problém, že nové spojení se pokouší rovněž o doručení souborů, které dříve nebyly odeslány. V důsledku toho nastala situace, kdy byl jeden soubor přenášen dvěma paralelními spojeními na kolektor. Tento efekt byl velice nežádoucí a vyžádal si nutnost okamžitého řešení.

První otázka je, zda má být vůbec přípustné, aby se z bufferu na kolektor otevíralo více paralelních spojení. Argumentem proti je skutečnost, že více spojení může snadněji zahltit linku a v důsledku může vést ke zpomalení přenosu. Na druhou stranu, pokud by bylo povoleno jen jedno spojení na kolektor, pak by docházelo ke zpožďování v doručení nejnovějších souborů, neboť by se doručily až po dokončení přenosu všech dříve nepřenesených souborů a tohle chování není také žádoucí. Kompromisem mezi těmito problémy byla zvolena možnost dvou paralelních spojení s tím omezením, že nově vzniklé spojení přenese pouze nejnovější soubor a nebude se pokoušet o doručení všech dříve neodeslaných souborů. Díky tomuto řešení je zaručeno, že všechny nové soubory jsou přeneseny okamžitě, je-li linka v pořádku.

Pokud jsou aktivní obě spojení, a dojde k výpadku linky pak se nezdaří přenos souboru ani v jednom spojení a dojde k ukončení pokusu o jejich doručení. V dalším cyklu dojde k přenosu nově vzniklého logu a k opětovnému pokusu doručit oba soubory, které se nepodařilo přenést v obou předchozích spojeních i všech následujících nepřenesených souborů.

7.4.2 Krátkodobý výpadek v době přenosu

Dalším provedeným testem byl test reakce na výpadek spojení během přenosu souboru. Popis testu v tabulce 7.3.

Název	Test dlouhodobého výpadku
Cíl	Ověřit, že po dlouhodobém výpadku budou v konečném čase doručeny všechny soubory na kolektor
Popis	Během činnosti (přibližně po 30 minutách) dojde k rozpojení linky mezi sondou a centrálním kolektorem na přibližně 4 hodiny (60 % doby celkového testu). Očekává se, že na konci testu budou všechna data dostupná na kolektoru.
Počáteční podmínky	1. Na bufferu nejsou žádná data k přenosu 2. Spojení mezi bufferem a kolektorem je funkční 3. Buffer přijímá data
Koncové podmínky	Jsou přenesena všechna data
Postup	1. Spustit buffer 2. Spustit přehrávání zdrojového souboru 3. Ověřit činnost 4. Přibližně po 30 minutách rozpojit linku mezi bufferem a kolektorem 5. Přibližně po 4 hodinách obnovit spojení mezi bufferem a kolektorem 6. Po celkové době testu 6 hodin zastavit test 7. Ověřit koncové podmínky
Závěr	Během tohoto testu bylo cílem přenést na kolektor stejná data, jako v předchozím případě, i při delší nedostupnosti spojení. Na konci testu byly všechny soubory na kolektoru a tento test byl splněn.

Tabulka 7.2: Dlouhodobý test

Název	Test krátkodobého výpadku během přenosu
Cíl	Ověřit stav, kdy výpadek nastane právě během přenosu souboru z bufferu na kolektor.
Popis	Během činnosti (po ověření, že data jsou na kolektor doručována) se počká na fázi, kdy dochází k doručení souboru a uprostřed této fáze ze provede krátkodobý výpadek (1 - 5 sekund) rozpojením linky. Očekává se, že na konci testu budou všechna data na kolektoru.
Počáteční podmínky	1. Na bufferu nejsou žádná data k přenosu 2. Spojení mezi bufferem a kolektorem je funkční 3. Buffer přijímá data
Koncové podmínky	Jsou přenesena všechna data
Postup	1. Spustit buffer 2. Spustit přehrávání zdrojového souboru 3. Ověřit činnost 4. Přibližně po 10 minutách vyčkat na fázi dalšího přenosu dat 5. V této fázi provést krátké rozpojení linky (1 - 5 sekund) 6. Po zapojení nechat běžet alespoň 10 minut 7. Ověřit koncové podmínky
Závěr	Všechna data, která byla exportérem odeslána byla po ukončení testu na kolektoru.

Tabulka 7.3: Test krátkodobého výpadku

Název	Test opakovaných výpadků
Cíl	Ověřit, že náhodně vyskytující se výpadky spojení nezpůsobí ztrátu dat
Popis	Během činnosti (po ověření, že data jsou na kolektor doručována) se provede několik opakujících se výpadků s různou délkou trvání.
Počáteční podmínky	1. Na bufferu nejsou žádná data k přenosu 2. Spojení mezi bufferem a kolektorem je funkční 3. Buffer přijímá data
Koncové podmínky	Jsou přenesena všechna data
Postup	1. Spustit buffer 2. Spustit přehrávání zdrojového souboru 3. Ověřit činnost 4. Počkat náhodně dlouhou dobu (1 sekunda až 5 minut) 5. Rozpojit linku na krátký okamžik (1 s až 1 minuta) 6. Opakovat alespoň 15 krát od bodu 4 7. Ověřit koncové podmínky
Závěr	Všechna data, která byla exportérem odeslána byla po ukončení testu na kolektoru.

Tabulka 7.4: Test krátkodobého výpadku

7.4.3 Opakované výpadky linky na krátké okamžiky

Aby byla sada testů při výpadku linky kompletní, byl doplněn i test chování při opakovaném vypadávání linky na různě dlouhé okamžiky. Během tohoto testu rovněž nebylo zjištěno nekorektní chování a navržené řešení je tedy možné označit za odolné vůči výpadkům (viz tabulka 7.4).

7.4.4 Testy při zatížené lince

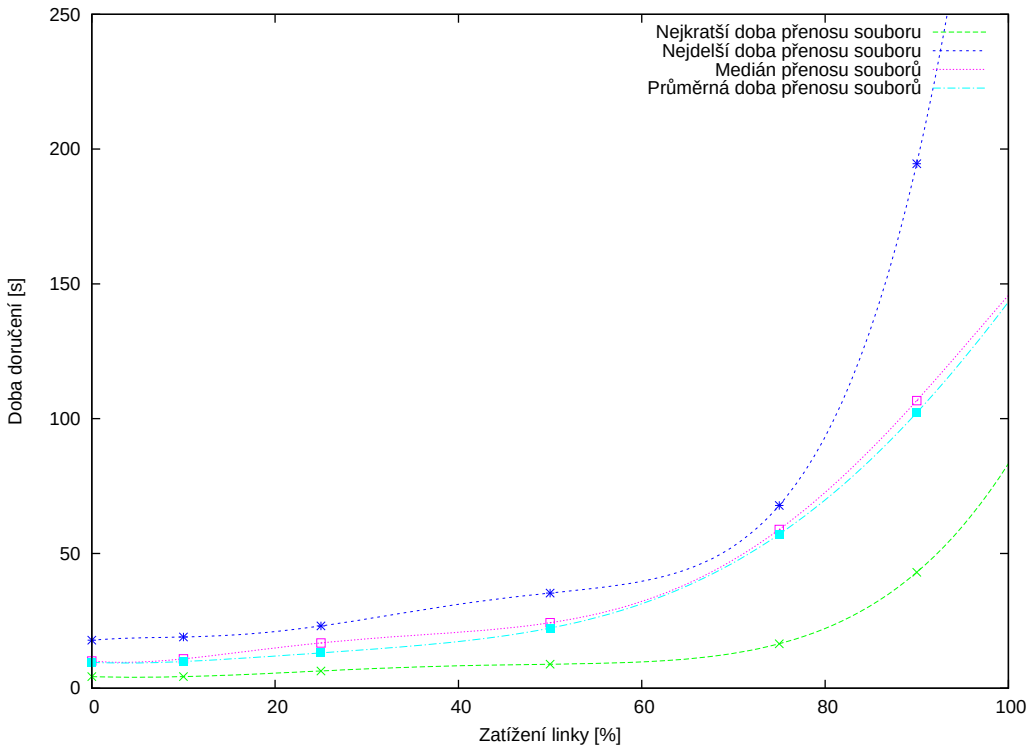
Po ověření správnosti předchozími testy byl proveden test pro porovnání doby přenosu při různě zatížené lince.

7.5 Spolehlivé doručení

Cílem projektu je zajistit spolehlivé doručení dat na kolektor. Výše realizované testy ověřily, že zvolené řešení garantuje doručení v případě jakýchkoliv problémů s linkou. Pro garanci spolehlivého doručení byly provedeny ještě testy, které simulovaly problém s kolektorem.

Vlastnost	**0**	**10**	**25**	**50**	**75**	**90**	**100**
Počet přenesených souborů	72	72	72	72	72	72	0
Nejkratší doba přenosu souboru	4,3 s	4,3 s	6,4 s	8,9 s	16,5 s	43,0 s	∞
Nejdelší doba přenosu souboru	17,8 s	19.0 s	23,1 s	35,3 s	67,8 s	194,6 s	∞
Medián přenosu souborů	10,1 s	10,9 s	16,8 s	24,3 s	58,9 s	106,7 s	∞
Průměrná doba přenosu souborů	9,6 s	9,9 s	13,1 s	22,3 s	57,1 s	102,2 s	∞

Tabulka 7.5: Charakteristiky přenosu při různém zatížení linky



Obrázek 7.7: Graf zobrazující změnu doby doručení souboru v závislosti na zatížení linky

Nedostupnost kolektoru je problém ekvivalentní výpadku linky a není tedy nutné analyzovat chování situace neběžícího SSH serveru. Nejčastější problém, který může na kolektoru nastat, je nedostatek volného místa na cílovém disku. Ověření odolnosti vůči zmíněnému problému bylo úspěšné.

Problém nedostatku diskového prostoru může teoreticky nastat i na bufferu. Z provedené analýzy bylo zjištěno, že délka výpadku by musela být mnohem delší než týden. Vzhledem k tomu, že NetFlow slouží k monitorování provozu sítě, lze očekávat, že takto dlouhý výpadek by byl odhalen mnohem dříve, než k takovéto situaci dojde.

7.6 Sbírání dat z více exportérů

Již v počátku testování byl objeven problém, že v případě, kdy je na sondě (bufferu) definováno více zdrojů, není chování zcela korektní. Dochází k přímé ztrátě dat. Problém spočívá v tom, že každý zdroj má definovanou svou *basedir*, která není součástí předávaného parametru. V důsledku toho jeden zdroj přepisuje v cíli ten druhý. Odchycení tohoto problému na

úrovni bufferu by znamenalo implementovat i serverovou část, která nedovolí přenos v podobném případě. Tento postup nevede k řešení tohoto problému. Řešení této komplikace bylo provedeno tak, že buffer vyžaduje ještě druhý parametr, kterým se předává identifikace zdroje. Je zajištěno, že každý zdroj je v kolektoru ukládán ve vlastním adresáři. Tím je zachován stejný přístup, jako na samotném kolektoru, pokud přijímá přímo NetFlow data.

7.7 Utajení přenášených dat

Žádný test pro testování, zda jsou data skutečně utajeny se neprováděl. Transport dat mezi exportérem a bufferem je realizován oddělenou linkou a není tedy možné tato data odposlechnout. Přenos dat je přes veřejnou síť prováděn protokolem SSH a jeho bezpečnost se neověřovala, neboť tento protokol je obecně uznáván za bezpečný.

7.8 Vyhodnocení

Testy provedené nad navrženým řešením ukázaly, že takto koncipovaný přístup dovoluje zajistit důvěrný a spolehlivý přenos NetFlow dat mezi exportérem a kolektorem. V případě, že se použije sonda založená na operačním systému Linux, není nutné realizovat buffer pomocí samostatného počítače. Samotnou sondu lze upravit tak, aby zajistila bezpečný přenos přes veřejnou síť na centrální kolektor.

Závěr

Cílem této práce bylo zaměřit se na analýzu transportních protokolů monitorovacích systémů pro sledování provozu na síti. V první části práce byly popsány tři protokoly sloužící k tomuto účelu a byla provedena analýza jejich vlastností a vlivu na síť.

V další části již protokol sFlow nebyl uvažován, neboť může sloužit výhradně k vzorkování sítě a cílem bylo využít protokol, který nabídne možnost sbírat data o všech tocích. Vliv na síť je v případě protokolu NetFlow a IPFIX shodný a rozhodování tedy ovlivňují pouze jejich vlastnosti. Z uvedeného pohledu je jasným vítězem protokol IPFIX, který má odstranit všechny nedostatky protokolu NetFlow. Bohužel vlastností, které se autoři IPFIX snaží do jeho standardu dostat, je mnoho a tento přístup byl již mnohokrát v historii důvodem nedokončení či neprosazení mnoha dobrých myšlenek. Tato skutečnost se v protokolu IPFIX zatím projevuje v jeho pomalém procesu standardizace, což v důsledku zpomaluje i zaváděni podpory v hardware. Důsledkem toho je protokol NetFlow vítězem pro další používání.

S ohledem na tyto skutečnosti byla další část této práce orientována na způsoby zabezpečení protokolu NetFlow. Výjimku tvořil rozbor SPoF monitorovacích systémů a monitorovaných sítí.

Z rozboru NetFlow protokolu vyplynula nutnost řešit zabezpečení důvěrnosti dat a spolehlivosti přenosu do vzdálené lokality skrze veřejnou síť, neboť implicitně ani jednu z těchto vlastností protokol není schopen zajistit sám o sobě. Protože protokol sFlow je zatížen stejnými nedostatky, posloužily jako odrazový můstek doporučení uvedena ve specifikaci tohoto protokolu. Přes několik iterací návrhu bylo nakonec přistoupeno k realizaci, která byla schopna nabídnout všechny požadované možnosti. Výsledné řešení umožňuje okamžité nasazení bez nutnosti složité instalace a konfigurace. Opakovaným testováním různých situací byla činnost tohoto řešení ověřena.

Současná situace kolem IPFIX není zcela vyjasněná. Není možné určit časový horizont, kdy by mohlo dojít k jeho masivnějšímu rozšíření, a tedy i podpoře v hardware. Navíc některé vlastnosti, které mohou zajistit spolehlivý a utajený přenos dat z exportéru na kolektor jsou v původním návrhu označeny za volitelné a jiné, které byly původně označené za povinné se nahrazují volitelnými. V důsledku této situace, je pravděpodobné, že při nasazení IPFIX bude nutné řešit stejné problémy, kterými se zabývá tato práce. Bude-li podpora protokolu IPFIX zahrnuta do balíku *nfdump* bude možno aplikovat získané řešení i na protokol IPFIX, neboť navržené řešení se stává z oddělených částí pro příjem dat a jejích přeposlání a není tedy fixováno na použitý protokol.

Literatura

[1] Introduction to Cisco IOS NetFlow - A Technical Overview.
URL `http://www.cisco.com`, Říjen 2007, Cisco Systems, White paper, Citováno 23. 3. 2010.

[2] Arkko, J.; Bradner, S.: IANA Allocation Guidelines for the Protocol Field.
URL `http://tools.ietf.org/rfc/rfc5237.txt`, Únor 2008.

[3] Boschi, E.; Mark, L.; Quittek, J.; aj.: IP Flow Information Export (IPFIX) Implementation Guidelines.
URL `http://tools.ietf.org/rfc/rfc5153.txt`, Duben 2008.

[4] B.Reese: Cisco's NetFlow vs. Inmon's sFlow: Which will prevail?
URL `http://www.networkworld.com/community/node/22667`, Květen 2007.

[5] Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information.
URL `http://tools.ietf.org/rfc/rfc5101.txt`, Leden 2008.

[6] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954.
URL `http://www.ietf.org/rfc/rfc3954.txt`, Říjen 2004.

[7] Cooper, D.; Santesson, S.; Farrell, S.; aj.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
URL `http://tools.ietf.org/rfc/rfc5280.txt`, Květen 2008.

[8] Farinacci, D.; Li, T.; Hanks, S.; aj.: Generic Routing Encapsulation (GRE).
URL `http://tools.ietf.org/rfc/rfc2784.txt`, Březen 2000.

[9] Grégr, M.: *Detekce a izolace útočníků pomocí záznamů NetFlow*. Fakulta informačních technologií Vysoké učení technické v Brně, 2009, diplomová práce.

[10] Haag, P.; Jändling, T.: NFDUMP.
URL `http://nfdump.sourceforge.net/`, citováno 23. 4. 2010.

[11] Hornig, C.: A Standard for the Transmission of IP Datagrams over Ethernet Networks.
URL `http://tools.ietf.org/rfc/rfc894.txt`, Duben 1984.

[12] Housley, R.; Ford, W.; Polk, W.; aj.: Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
URL `http://tools.ietf.org/rfc/rfc2459.txt`, Leden 1999.

[13] Kent, S.; Atkinson, R.: Security Architecture for the Internet Protocol.
URL http://tools.ietf.org/rfc/rfc2401.txt, Listopad 1998.

[14] Kohler, P.: NetFlow for Accounting, Analysis and Attack.
URL http://www.cisco.com/networkers/nw04, 2004, Networkes, NMS-2032, Citováno 30. 3. 2010.

[15] Menezes, A.; van Oorschot, P.; Vanstone, S.: *Handbook of Applied Cryptography*. CRC Press, October 1996, 816 pages, ISBN 0-8493-8523-7.

[16] Phaal, P.; Panchen, S.; McKee, N.: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks.
URL http://tools.ietf.org/rfc/rfc3176.txt, Září 2001.

[17] Quittek, J.; Zseby, T.; Claise, B.; aj.: Requirements for IP Flow Information Export (IPFIX). RFC 3917.
URL http://tools.ietf.org/rfc/rfc3917.txt, Říjen 2004.

[18] Rescorla, E.; Modadugu, N.: Datagram Transport Layer Security.
URL http://tools.ietf.org/rfc/rfc4347.txt, Duben 2006.

[19] Rosen, E.; Viswanathan, A.; Callon, R.: Multiprotocol Label Switching Architecture.
URL http://tools.ietf.org/rfc/rfc3031.txt, Leden 2001.

[20] Stewart, R.; Ramalho, M.; Xie, Q.; aj.: Stream Control Transmission Protocol (SCTP) Partial Reliability Extension.
URL http://tools.ietf.org/rfc/rfc3758.txt, Květen 2004.

[21] Thayer, R.; Doraswamy, N.; Glenn, R.: IP Security Document Roadmap.
URL http://tools.ietf.org/rfc/rfc2411.txt, Listopad 1998.

[22] WWW stránky: Caligare :: Netflow monitoring software.
URL http://netflow.caligare.com, citováno 2. 5. 2010.

[23] WWW stránky: INVEA-TECH - High-Speed Networking and FPGA Solutions.
URL http://www.invea.cz/, citováno 14. 5. 2010.

[24] WWW stránky: Netflow – Wikipedia, the free encyclopedia.
URL http://en.wikipedia.org/wiki/Netflow, citováno 23. 4. 2010.

Seznam použitých zkratek

Zkrakta	**Význam**
AH	Authentication Header
ATM	Asynchronous Transfer Mode
CVIS	Centrum výpočetních a informačních služeb
DTLS	Datagram Transport Layer Security
ESP	Encapsulating Security Payload
FIN	Příznak paketu ukončující spojení
GRE	Generic Routing Encapsulation
HA	High Availability
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPSec	Internet Protocol Security
IPv6	Internet Protocol version 6
ISO/OSI	International Standards Organization / Open System Interconnection
LDAP	Lightweight Directory Access Protocol
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
PR	Partially Reliable
PR-SCTP	Partially Reliable for Stream Control Transmission Protocol
QoS	Quality of Service
RFC	Request for Comments
RST	Příznak paketu resetující spojení
SCTP	Stream Control Transmission Protocol
SNMP	Simple Network Management Protocol
SPAN	Switch Port Analyzer
SPoF	Single Point of Failure
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TLV	Type Length Value
ToS	Type of Service
UDP	User Datagram Protocol
UTC	Universal Time Coordinated
VLAN	Virtual Local Area Network
VPN	Virtual private network
VUT	Vysoké učení technické

Seznam obrázků

Seznam tabulek